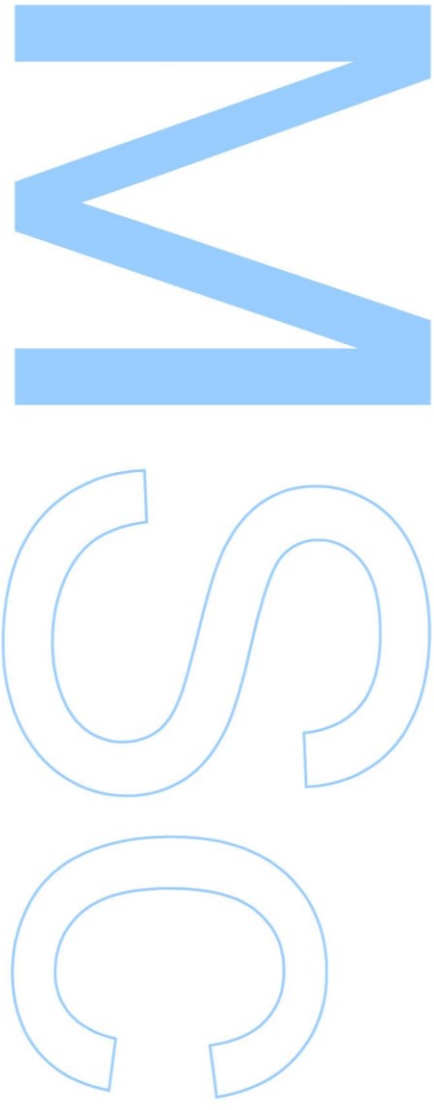
# Time Series Analysis based on Complex Networks

Vanessa Alexandra Freitas da Silva
Master's degree in Networks and Informatics Systems Engineering
Computer Science Department
2018

**Supervisor**
Fernando Manuel Augusto da Silva, Full Professor, Faculty of Sciences, University of Porto

**Co-supervisor**
Pedro Manuel Pinto Ribeiro, Assistant Professor, Faculty of Sciences, University of Porto

**Co-supervisor**
Maria Eduarda da Rocha Pinto Augusto da Silva, Associate Professor, Faculty of Economics, University of Porto



U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____

# Abstract

Time series data are ubiquitous in domains as diverse as climate studies, economics and health care. Mining interesting features from these data is therefore a crucial task with an inherent multidisciplinary impact. Time series analysis is a well established research field with a vast array of available methodologies. Due to recent technological advances, ranging from remote sensing to wearables and social sensing, there is a tremendous increase in time series data that is available and requires analysis.

Complex networks can also describe a wide range of systems in nature and society and their analysis has been receiving increasing interest from the research community. The impact has been so big that has led to the emergence of the new field of Network Science, and there exists a vast set of topological graph measurements available, an established set of problems and a large track record of successful applications.

At first sight, time series and complex networks do not seem to be related, but there have been recent advances that try to leverage the knowledge of both worlds. The classical approaches to time series analysis present severe limitations when analysing multidimensional sets of time series. A recent and very promising conceptual approach relies on mapping the time series to complex networks, where the vast arsenal of network science methodologies can help to gain new insights into the mapped time series.

This thesis main goal focuses precisely on contributing to time series analysis based on complex networks. We first give an overview the main concepts of both areas and we survey the literature on mappings that able to transform time series into complex networks that preserve some of its characteristics. These mappings are based on concepts such as correlation, phase space reconstruction, recurrence analysis, visibility or transition probabilities.

We then perform a systematic network based characterization of a large set of linear and nonlinear time series models using topological properties of the constructed networks using the visibility and transition probabilities concepts. We show that different mappings and different topological metrics capture different characteristics, complementing each other and providing more information when combined than simply being considered by themselves.

Finally, a novel time series clustering methodology is proposed. The approach advocates combining mappings and network metrics into vast and rich feature sets. Coupling this with dimensionality reduction techniques and classical clustering algorithms, and when applied to synthetic data, our method is able to distinguish linear from nonlinear, stationary from nonstationary, pseudo periodic from nonperiodic and autocorrelated from uncorrelated data. Moreover, clustering results in real data sets are competitive with other more classical approaches described in the literature, indicating the potential of network based analysis to contribute to open problems in time series.

**Keywords:** Time Series, Complex Networks, Characterization and Clustering, Topological Features, Nonparametric Methods

# Resumo

Dados de séries temporais são omnipresentes em domínios tão diversos como estudos climáticos, economia e assistência médica. A obtenção de características interessantes a partir destes dados é, portanto, uma tarefa crucial com um impacto multidisciplinar inerente. A análise de séries temporais é um campo de pesquisa bem estabelecido com um vasto conjunto de metodologias disponíveis. Devido aos recentes avanços tecnológicos, desde sensores remoto a wearables e monitorização social, há um tremendo aumento nos dados de séries temporais disponíveis e necessidade de análise.

As redes complexas também podem descrever uma ampla gama de sistemas naturais e sociais e a sua análise recebeu um crescente interesse da comunidade de pesquisa. O impacto foi tão grande que levou a um novo campo de *Network Science*, e existe um vasto conjunto de medidas topológicas de grafos disponíveis, um conjunto estabelecido de problemas e um grande histórico de aplicações bem-sucedidas.

À primeira vista, séries temporais e redes complexas não parece estarem relacionadas, mas houve avanços recentes que tentam alavancar o conhecimento de ambos os mundos. As abordagens clássicas para análise de séries temporais apresentam grandes limitações ao analisar conjuntos multidimensionais de séries temporais. Uma recente abordagem conceptual e bastante promissora depende do mapeamento de séries temporais para redes complexas, onde o vasto arsenal de metodologias de *Network Science* pode ajudar a obter novas descobertas sobre as séries temporais mapeadas.

Esta tese tem precisamente como objetivo contribuir para a análise de séries temporais usando redes complexas. Primeiro, começamos por dar uma visão geral dos principais conceitos de ambas as áreas e descrevemos o estado da arte dos mapeamentos que transformam séries temporais em redes complexas preservando algumas das suas características. Estes mapeamentos são baseados em conceitos como correlação, reconstrução do espaço de fases, análise de recorrência, visibilidade ou probabilidades de transição.

Realizamos então uma caracterização sistemática baseada em redes de um grande conjunto de modelos de séries temporais lineares e não lineares usando propriedades topológicas de redes construídas usando os conceitos de visibilidade e de probabilidades de transição. Mostramos que diferentes mapeamentos e diferentes métricas topológicas capturam diferentes características, complementando-se e fornecendo mais informações quando combinadas do que simplesmente

consideradas por si só.

Finalmente, propomos um novo método de *clustering* de séries temporais. A nossa proposta defende a combinação de diferentes mapeamentos e métricas num vasto e rico conjunto de características. Juntando isto com técnicas de redução da dimensionalidade e algoritmos clássicos de *clustering*, e quando aplicado a conjuntos de dados sintéticos, o nosso método distingue dados lineares de não-lineares, estacionários de não-estacionários, pseudo-periódicos de não-periódicos e correlacionados de não-correlacionados. Além disso, os resultados de *clustering* com séries reais são competitivos com outras abordagens mais clássicas da literatura e indicam o potencial da análise baseada em redes para contribuir para problemas abertos em séries temporais.

**Palavras-chave:** Séries temporais, Redes complexas, Caracterização e *Clustering*, Medidas topológicas, Métodos não paramétricos

# Acknowledgements

First of all, I would like to thank my three advisors, Fernando Silva, Pedro Ribeiro and Maria Eduarda Silva. They were always available when I needed them, providing guidance, support, patience, and motivation. Their collaboration was essential and we always had interesting and fruitful discussions on the emerging ideas on this subject. To Professor Fernando Silva and Professor Pedro Ribeiro wanted to make a special word of thanks for the motivation and all the advices given over the past two years. To Professor Maria Eduarda Silva, I would especially like to thank you for having agreed to work with us on this project and for all help and availability.

I would like to thank INESC TEC for its support in this Master's program, offering me the opportunity to learn and work more.

Finally, I would like to thank all my friends and family, that have always supported me and helped me maintain the balance between my academic and personal life. Especially my love, Ricardo, for his motivation, for his advice and for encouraging me in the most exhausting moments. And especially to my parents and two brothers that have always supported my choices and have always provided an atmosphere of joy and tranquility.

# Contents

# List of Tables

# List of Figures

# Acronyms

**ACF** Autocorrelation Function

**ARMA** Autoregressive Moving Average

**AR** Autoregressive

**MA** Moving Average

**ARIMA** Autoregressive Integrated Moving Average

**ARFIMA** Autoregressive Fractionally Integrated Moving Average

**SETAR** Self-Exciting Threshold Autoregressive

**HMM** Hidden Markov Models

**INAR** Integer-Valued Autoregressive

**ARCH** Autoregressive Conditional Heteroskedasticity

**GARCH** Generalized Autoregressive Conditional Heteroskedasticity

**EGARCH** Exponential Generalized Autoregressive Conditional Heteroskedasticity

**GJR-GARCH** Glosten-Jagannathan-Runkle Generalized Autoregressive Conditional Heteroskedasticity

**NVG** Natural Visibility Graph

**HVG** Horizontal Visibility Graph

**PNVG** Parametric Natural Visibility Graph

**WVG** Weighted Visibility Graph

**EEG** Electroencephalogram

**LPVG** Limited Penetrable Visibility Graph

**DVG** Diference Visibility Graphs

**MLPHVG** Multiscale Limited Penetrable Horizontal Visibility Graph

**QG** Quantile Graph

**PCA** Principal Component Analysis

**PC** Principal Component

**t-SNE** t-Distributed Stochastic Neighbor Embedding

# Chapter 1

# Introduction

A time series is a collection of observations indexed in time. The main purpose of time series analysis is to develop mathematical models that provide plausible descriptions of the characteristics of the data with a view to forecasting, simulation and control [9]. This type of analysis is used in a variety of important domains, such as climate studies [45], health monitoring [33] and financial data analysis [61]. The methods and the associate theory for univariate time series analysis are well developed and understood and a plethora of models exist to describe the behaviour of particular series, ranging from the linear ARIMA models for stationary time series to nonlinear GARCH type models for modelling conditional volatility and widely applied in financial data [34, 56].

We can say that time series analysis has three main objectives, namely description, explanation and forecast. The description is very important tasks and is the first step in the analysis of a time series, it consists of plotting the data and obtaining simple descriptive measures (e.g. looking for trends, seasonal variations, structural changes and calculate mean, variance and autocorrelation) to describe the important characteristics of the time series patterns [56].

Complex networks describe a wide range of systems in nature and society and their analysis has been receiving increasing interest from the research community [46]. The impact has been so big that has led to the emergence of the new field of Network Science [6]. They present substantial non-trivial topological features and there exists a vast set of topological graph measurements available [15], an established set of problems such as community detection [23] or link prediction [41], and a large track record of successful application of complex network methodologies to different fields [16].

## 1.1 Motivation

Motivated by the success of complex network methodologies and with the objective of acquiring new and better methods for the analysis of time series, several network-based time series analysis approaches have been recently proposed, based on mapping time series to the network domain.

The mappings proposed in the literature are based on concepts such as correlation [70], phase space reconstruction [26], recurrence analysis [43], visibility [36] or transition probabilities [11]. They result in networks that capture the structural properties of the series. For instance, periodic series are represented by regular networks, random series by random networks and chaotic series map to scale-free networks. Some mappings result in networks that have as many nodes as the number of observations in the time series, but others, such as a quantile based mapping [11], allow to reduce the dimensionality of the series while preserving the characteristics of the time dynamics. Network-based time series analysis techniques have been showing promising results and have been successful in the description, classification and clustering of time series of real datasets. Examples of this include automatic classification of sleep stages [71], characterizing the dynamics of human heartbeat [54], distinguishing healthy from non-healthy electroencephalographic (EEG) series [10] and analyzing seismic signals [59].

Clustering analysis is prevalent in any discipline that involves data analysis, and time series analysis is no exception [49]. Grouping time series into sets with similar features is also an intrinsic activity to facilitate the handling and the organization of the enormous amount of information that we can capture.

In this thesis we propose a new approach to cluster time series models through non-supervised learning. This approach consists of constructing three types of graphs for each of the simulated time series and calculating their global metrics for these graphs, namely, average degree, average path length, number of communities, cluster coefficient and modularity.

We performed a comparative analysis of the metrics obtained, as well as an exhaustive study on the clustering analysis using sets of these metrics, in order to show that different mapping concepts complement each other. By using at the same time features from different mappings, we obtain a richer and more powerful characterization, improving the results when comparing with simply using a single mapping concept, as is common on the literature.

Topologically similar networks are thus likely to represent systems with functional similarities, while networks with different topological properties represent different systems. This is indeed what we observed and will show in this work, since different time series models (but some with common properties) are mapped to networks with different topological metrics.

Other problems of time series analysis are, for example, the lack of model identification or flat likelihood functions and the large number of parameters (e.g. VAR models are well known to suffer the curse of dimensionality [63]), and validity of a parametric model for a large set of real data over a long period of time is always questionable. Consequently, new paradigms are being developed for working with multidimensional time series data in the era of big data. Some of these new paradigms rely on dimensionality reduction (for example, via reduced-rank structure, structural indices, scalar component models and canonical correlation analysis [21]).

## 1.2 Goal and Contributions

As mentioned, our goals roam around the problem of analyzing time series based on complex networks. To ease their presentation, we divided our work into two parts: one in relation to the topological measures that best characterize a time series and another in relation to the power of clustering of time series based on these measures.

The key question is whether the associated graph inherits some structure of the time series and consequently if the process that generated the time series can be characterized by the use of graph related concepts.

The main contributions of the work described in this thesis are the following:

- We propose the joint usage of two different time series to complex networks mapping concepts (visibility and transition probability), in order to show that different mappings capture different information from the time series, leading to more information being captured from them when combined. To the best of our knowledge, no other existing work does this. Previous work only takes into account one possible mapping concept, even if it modifies it for a specific task.
- We choose a set of five different global topological network metrics and we perform a detailed and thorough analysis of what time series characteristics are these metrics capturing when applied to both mapping concepts we proposed to use. We also take into account different time series lengths, to better understand if our conclusions are general or depend on this. As far as we know, no other work does a similar complete characterization.
- Our extensive empirical tests use a large set of time series models of various sizes, both linear and nonlinear, to explain their differences based on complex networks. By contrast, the existing works are focused on smaller and more commonly studied sets of time series types.
- We propose a powerful time series clustering method based on the usage of a very rich feature set constituted by all studied topological metrics for several different networks created using our proposed mapping concepts. We provide an extensive study of the discriminative power of our approach on a large set of time series models, showing both its validity and accuracy. The scale of our study is unmatched by any other previous work that we know of, both on the quantity of mapped networks and on the quantity of time series models used.

## 1.3 Organization

This thesis is structured into six major chapters. A brief description of each one of them is now provided.

**Chapter 1 - Introduction.** Provides a brief introduction to the research area, the main motivation, goals and contributions, and the organization of the thesis.

**Chapter 2 - Basic Concepts.** Introduces a common time series and complex networks terminology that will be used throughout the thesis and uses it to formalize the problem being tackled. It also provides a brief description of the time series models that we will analyze as well as description of the measures used to extract information from the time series and the complex networks.

**Chapter 3 - From Time Series to Complex Networks.** Introduces the main methods of time series to complex networks mapping, as well as a brief description of its main results and variations explored in the literature.

**Chapter 4 - Characterization of Time Series using Topological Features.** Performs a detailed description of the characteristics of several sets of time series using global topological metrics extracted from complex networks generated through methods of visibility and of transition probability.

**Chapter 5 - Clustering using Topological Features.** Performs an exhaustive clustering analysis of a large set of simulated time series models based on network measures. It also performs a comparison of the proposed approach with a approach of time series area on a real dataset.

**Chapter 6 - Conclusions.** Discusses the research done, summarizing the contributions made and gives directions for future work.

# Chapter 2

# Basic Concepts

In this chapter we present the background concepts on time series and complex networks and establish the notation necessary for the remaing of the thesis. We describe the main time series models, the main types of complex networks and some topological metrics of interest.

## 2.1 Time Series

A time series $Y_T = (y_1, \ldots, y_T)$ is a set of observations collected at different points in time. The main characteristic of a time series is the serial dependence between the observations which restricts the applicability of many conventional statistical models traditionally dependent of the assumption of independent and identically distributed observations. Time series analysis refers to the collection of procedures developed to systematically solve the statistical problems posed by the time correlations. Thus, the main purpose of the analysis of a time series is to develop mathematical models that provide plausible descriptions of the characteristics of the data with a view to forecasting, simulation and control [9].

### 2.1.1 Definitions

Formally, a time series is a realisation of a stochastic process which is a collection of random variables $\{y_t\}$ indexed by $t$. Here we will consider $t$ discrete and varying in the integers, $t = 0, \pm 1, \pm 2, \ldots$. From this definition, it follows that a complete description of a time series $y_{t_1}, \ldots, y_{t_n}$ is provided by the joint distribution function $F(c_1, \ldots, c_n) = P(y_{t_1} \leq c_1, \ldots, y_{t_n} \leq c_n)$. This joint distribution can be easily obtained only in the particular case of jointly Gaussian random variables and, even so, under the assumption of some regularity of the time series over time. Thus, time series analysis relies in a set of concepts and measures that capture the essential feature of the data, the serial correlation.

**Mean Function** The mean function is defined as

$$\mu_{yt} = E(y_t) \tag{2.1}$$

provided it exists, where $E$ denotes the usual expected value operator.

**Autocovariance and Autocorrelation Functions** The *autocovariance function* is defined as the second moment product

$$\gamma_y(s,t) = cov(y_s, y_t) = E[(y_s - \mu_s)(y_t - \mu_t)] \tag{2.2}$$

for all $s$ and $t$.

We should note that $\gamma_y(s,t) = \gamma_y(t,s)$ for all time points $s$ and $t$. The autocovariance measures the linear dependence between two points on the same series observed at different times. It is clear that, for $s = t$, the autocovariance reduces to the (assumed finite) variance, because

$$\gamma_y(t,t) = E[(y_t - \mu_t)^2] = var(y_t) \tag{2.3}$$

By traditional statistics if $\gamma_y(s,t) = 0$, $y_s$ and $y_t$ are not linearly related, but there still may be some dependence structure between them. However, if $y_s$ and $y_t$ are bivariate normal then this ensures their independence [56].

The *autocorrelation function* (ACF) is defined as

$$\rho_y(s,t) = \frac{\gamma_y(s,t)}{\sqrt{\gamma_y(s,s)\gamma_y(t,t)}} \tag{2.4}$$

The ACF measures the linear predictability of the series at time $t$, say $x_t$, using only the value $x_s$. Its value must lie in the range $[-1, 1]$, with 1 indicating perfect correlation and $-1$ indicating perfect anticorrelation. For example, if we can predict $y_t$ *perfectly* from $y_s$ through a linear relationship, $y_t = \beta_0 + \beta_1 y_s$, then the correlation will be 1 when $\beta_1 > 0$, and $-1$ when $\beta_1 < 0$. Hence, we have a rough measure of the ability to forecast the series at time $t$ from the value at time $s$ [56].

**Stationarity** In time series the notion of regularity referred to above is defined using the concept of stationarity. Technically, there are two forms of stationarity, strict (or strong) stationarity and weak (or wide or second order) stationarity.

A *strictly stationary* time series is one for which the probabilistic behavior of every collection of values

$$\{y_{t_1}, y_{t_2}, \ldots, y_{t_k}\}$$

is identical to that of the time shifted set

$$\{y_{t_1+h}, y_{t_2+h}, \ldots, y_{t_k+h}\}.$$

That is,

$$P\{y_{t_1} \leq c_1, \ldots, y_{t_k} \leq c_k\} = P\{y_{t_1+h} \leq c_1, \ldots, y_{t_k+h} \leq c_k\} \tag{2.5}$$

for all $k = 1, 2, \ldots$, all time points $t_1, t_2, \ldots, t_k$, all numbers $c_1, c_2, \ldots, c_k$, and all time shifts $h = 0, \pm 1, \pm 2, \ldots$.

This form is is too strong for most applications, and it is difficult to assess strict stationarity from a single dataset. Rather than imposing conditions on all possible distributions of a time series, the weak stationarity imposes conditions only on the first two moments of the series.

Thus, a *weakly stationary* time series is a finite variance process such that

(i) the mean value function, $\mu_t$, is constant and does not depend on time $t$, and

(ii) the autocovariance function, $\gamma_y(s, t)$, depends on $s$ and $t$ only through their difference $|s-t|$.

By the second condition, ii, we can simplify the autocovariance function. Let $s = t + h$, where $h$ represents the time shift or lag. Then

$$\gamma_y(t + h, t) = cov(y_{t+h}, y_t) = cov(y_h, y_0) = \gamma_y(h, 0) \tag{2.6}$$

because the time difference between times $t + h$ and $t$ is the same as the time difference between times $h$ and 0. Thus, the autocovariance function of a stationary time series does not depend on the time argument t, and is written as

$$\gamma_y(h) = cov(y_{t+h}, y_t) = E[(x_{t+h} - \mu)(x_t - \mu)]. \tag{2.7}$$

Consequently, the autocorrelation function of a stationary time series will be written as

$$\rho_y(h) = \frac{\gamma_y(t + h, t)}{\sqrt{\gamma_y(t + h, t + h)\gamma_y(t, t)}} = \frac{\gamma_y(h)}{\gamma_y(0)} \tag{2.8}$$

We say that a time series is stationary when its statistical characteristics (mean, variance) are constant over time, that is, its data oscillate around a constant mean with the variance of the fluctuations remaining essentially the same. Moreover, stationarity implies that correlation between observations depends only on the time lag between the observations.

### 2.1.2 Time Series Description

Stationarity is essential in the analysis of time series since most of the methods and models require that the series are stationary. However, many of the time series of interest are inherently nonstationary, exhibiting trend, cycles, seasonal patterns, and/or other nonstationary behaviours. The traditional approach to analyse the data under these circumstances is to decompose the time series in *trend*, *cycle*, *seasonality* and *random components*. The trend component indicates the

long term behavior of the series. The cycle is characterized by smooth and repeated oscillations of rise and fall in the time series around the trend. The seasonal component corresponds to the oscillations of ascent and descent that occur with a fixed period within a year: they are usually related to the seasons of the year. The seasonal component presents predictable movements at regular time intervals, as opposed to cyclical component which is quite irregular. The random component represents all the other effects resulting from a multiplicity of factors and of unpredictable nature. It is usually a stationary time series and the machinery for dealing with stationary time series may be applied.

We can distinguish three tasks that the time series analysis has on the data under analysis, they are, description, explanation (or modeling) and forecasting.

**Description** The description is the first step in the analysis of a time series, it consists of plotting the data and obtaining simple descriptive measures (for example, looking for trends, seasonal variations, structural changes and calculate mean, variance and autocorrelation) to describe the important characteristics of the standard of the series. For example, if we look at figure 2.1, which represents the global mean land–ocean temperature index from 1880 to 2009, we can see that there is an apparent upward trend in the series during the latter part of the twentieth century, a leveling around 1935 and a fairly steep upward trend around 1970 [56].

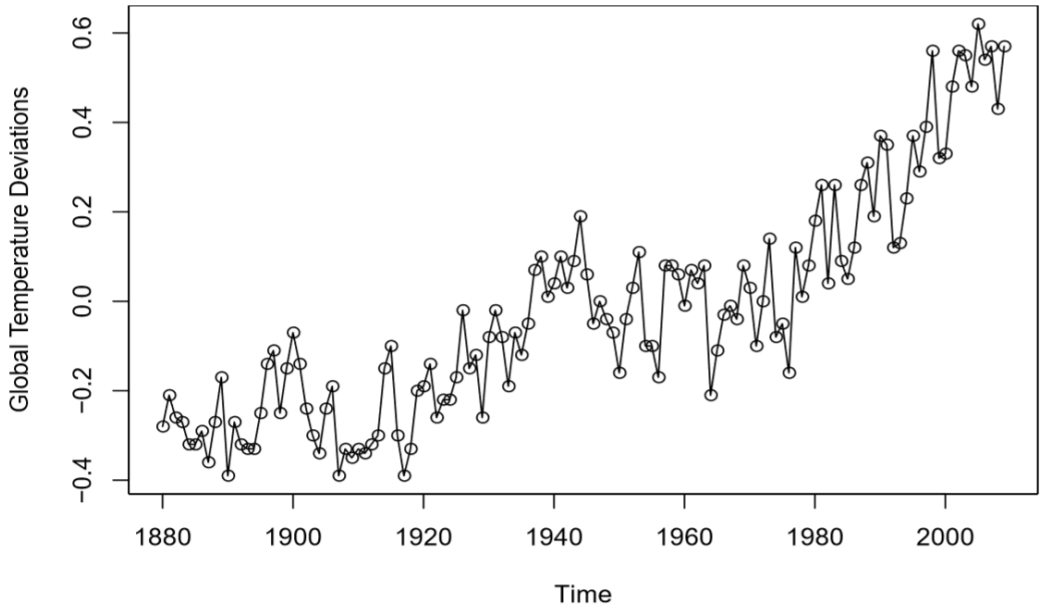



Figure 2.1: Yearly average global temperature deviations (1880–2009) in degrees celsius [56].

Before attempting to model and forecast a time series, it is desirable to have a preliminary idea of the data in order to obtain some of its main properties (in the example, the trend). This is of great use as it can help in choosing the most appropriate set of models (from a wide range available) and even their parameters.

**Explanation/Modelling** The modeling (or explanation) of the stochastic process is one of the most important tasks and consists in obtaining a mathematical model that represents process dynamics. This encompasses three stages: identification, estimation and diagnostics that are iterated until the serial correlation has been accounted for [9]. This task is very complex and is influenced by several factors, *a priori* knowledge of the process (linear, nonlinear, ...), model properties (degree of complexity), error measures to be minimized, and presence of noise [2].

**Forecast** One of the most important purposes of time series analysis is forecasting, which consists of forecast future values of a given time series observed on the basis of past and present values, and is of great interest in several fields [9].

### 2.1.3 Time Series Models

There are many time series models available in the literature. The models are classified into two main types: linear and nonlinear. Here we describe the most popular and useful models for time series which we will considered throughout this work.

The simplest time series process is the purely random process or *white noise*, henceforth denoted by $\epsilon_t$. A white noise is a sequence of independent and identically distributed (i.i.d.) random variables with zero mean and constant variance, $\sigma_\epsilon^2$. A particular case is the Gaussian white noise, where $\epsilon_t$ are independent normal random variables [56]. In figure 2.2 we present an example of a time series of length 1000 generated by a process white noise ($\epsilon_t \sim N(0,1)$).

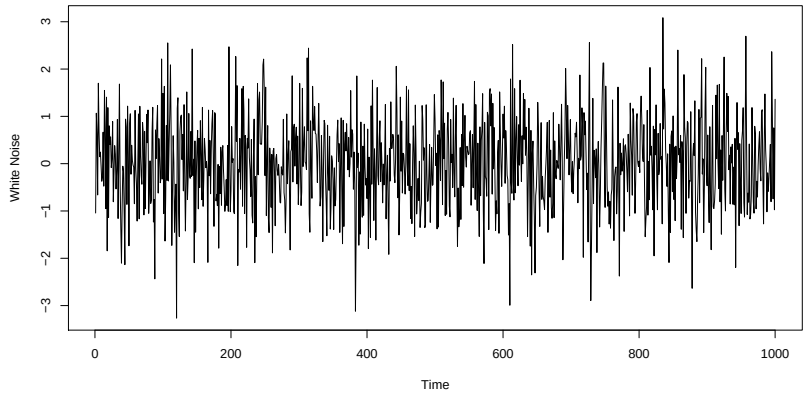



Figure 2.2: A representation from a white noise process $\epsilon_t \sim N(0,1)$.

White noise is the building block of all the other time series models and reflects information that is not directly observable in the time series. For this reason it is often called an innovations process [22].

#### 2.1.3.1 Linear Models

According to Cryer and Chan [17], linear time series models are essentially models for which the conditional mean is a linear function of past values of the time series. The most popular class of linear time series models are the autoregressive moving average (ARMA) models, which include purely autoregressive (AR) and moving-average (MA) models. A particularly useful model class are the so-called autoregressive integrated moving average (ARIMA) models, which includes stationary ARMA processes as a subclass [22].

**AutoRegressive Model** We define $y_t$ as an autoregressive process of order $p$, denoted by AR($p$), if it satisfies the following equation [56]:

$$\begin{aligned} y_t &= \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t \\ (1 - \sum_{i=1}^{p} \phi_i B^i) y_t &= \epsilon_t \\ \Phi(B) y_t &= \epsilon_t \end{aligned} \tag{2.9}$$

where $B$ represents the backshift operator, $By_t = y_{t-1}$, $p$ is the number of autoregressive terms and $\epsilon_t$ is a white noise process. This model explicitly specifies a linear relationship between the current and past values as suggested by its name. An AR($p$) process is stationary if the zeros $z_1, \ldots, z_p$ of the autoregressive polynomial $\Phi(z) = 1 - \phi_1 z - \ldots - \phi_p z^p \neq 0$ for $|z| \leq 1$. The autocorrelation of an AR processes decays exponentially to zero, $\rho(k) = \sum_{i=1}^{p} \phi_i \rho(k-i),\ k \geq p$ meaning that far away observations are almost uncorrelated.

In this work we will study four particular AR models: three AR(1) models with $\phi_1 = \{-0.5, 0.5, 0.9\}$ and one AR(2) model with $\phi_1 = 1.5$ and $\phi_2 = -0.75$. A more detailed explanation of why we choose this particular set of parameters, both for this and the following models, is given in section 4.1. For the sake of better visual understanding, we include in this chapter figures depicting simulated instances of length $T = 1000$ from all of the used models. In particular, figures 2.3 to 2.6 depict examples of all the four described AR models.

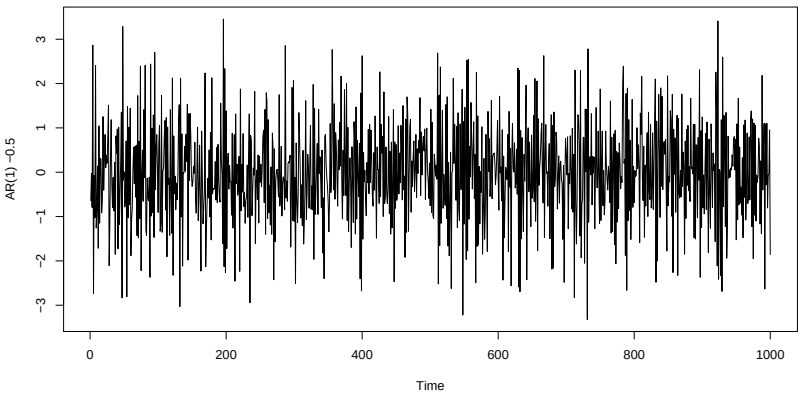



Figure 2.3: A representation from a AR(1) model $y_t = -0.5y_{t-1} + \epsilon_t$.

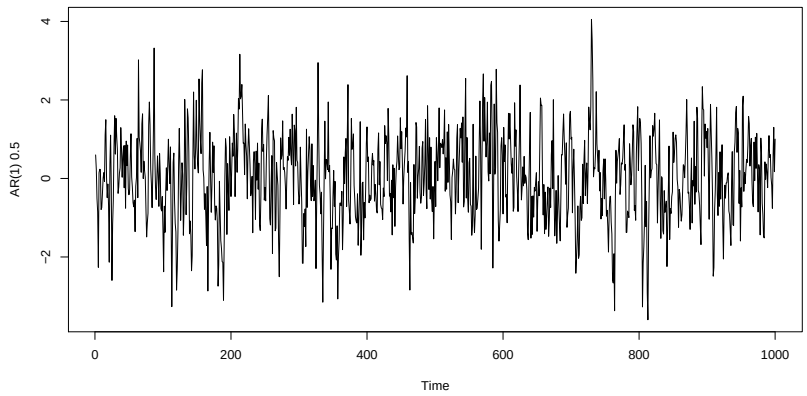



Figure 2.4: A representation from a AR(1) model $y_t = 0.5y_{t-1} + \epsilon_t$.

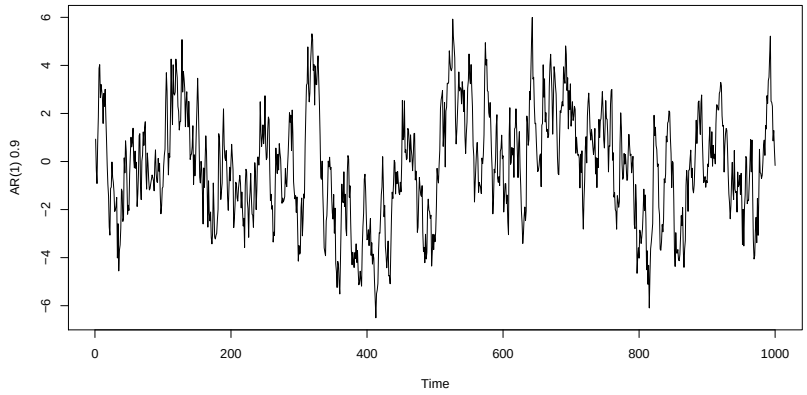



Figure 2.5: A representation from a AR(1) model $y_t = 0.9y_{t-1} + \epsilon_t$.

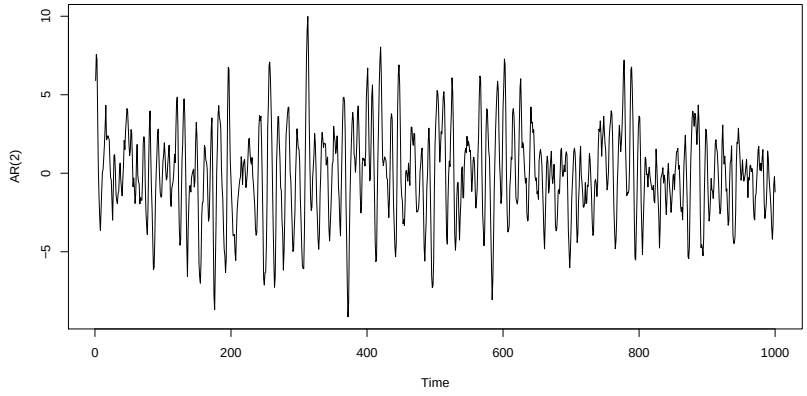



Figure 2.6: A representation from a AR(2) model $y_t = 1.5y_{t-1} - 0.75y_{t-2} + \epsilon_t$.

**AutoRegressive Integrated Moving Average** An autoregressive moving average, ARMA, process combines AR proceses and Moving Average, MA, processes which consist of a linear combination of i.i.d random variables (white noise) [56]. Thus, $y_t$ is an ARMA process of order $(p, q)$ if it satisfies the equation

$$\begin{aligned} y_t &= \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \\ \left(1 - \sum_{i=1}^{p} \phi_i B^i\right) y_t &= (1 + \sum_{i=1}^{q} \theta_i B^i)\epsilon_t \\ \Phi(B) y_t &= \Theta(B)\epsilon_t \end{aligned} \tag{2.10}$$

where the white noise $\epsilon_t$ is usually a Gaussian process, $\phi_i$, $i = 1, \ldots, p$ are constants such that $\Phi(z) = 1 - \sum_{i=1}^{p} \phi_i z^i \neq 0$ for $|z| \leq 1$ and $\theta_i$, $i = 1, \ldots, q$ are constants such that $\Theta(z) = 1 + \sum_{i=1}^{q} \theta_i z^i \neq 0$ for $|z| \leq 1$. Under these conditions the ARMA$(p, q)$ is stationary and invertible. The latter condition is important for forecasting purposes. The autocorrelation function of the ARMA$(p, q)$ also satisfies $\rho(k) = \sum_{i=1}^{p} \phi_i \rho(k - i),\ k \geq \max(p, q)$ since the MA$(q)$ process is uncorrelated for lags greater than $q$.

Now, assume that you have a nonstationary time series, $x_t$ but whose $d$th-difference $y_t = \nabla^d x_t$ is a stationary ARMA$(p, q)$ process. Then $x_t$ is said an AutoRegressive Integrated Moving Average, ARIMA$(p, q)$, process and satisfies the following equation:

$$\begin{aligned} \Phi(B) y_t &= \Theta(B)\epsilon_t \\ \Phi(B) y_t (1 - B)^d x_t &= \Theta(B)\epsilon_t \end{aligned} \tag{2.11}$$

Thus ARIMA$(p, d, q)$ models generalize ARMA$(p, q)$ models and have been quite successful in modeling non stationary time series.

Figure 2.7 represents an example of a time series of length 1000 generated by an ARIMA(1,1,0) model that we will studv in this work.

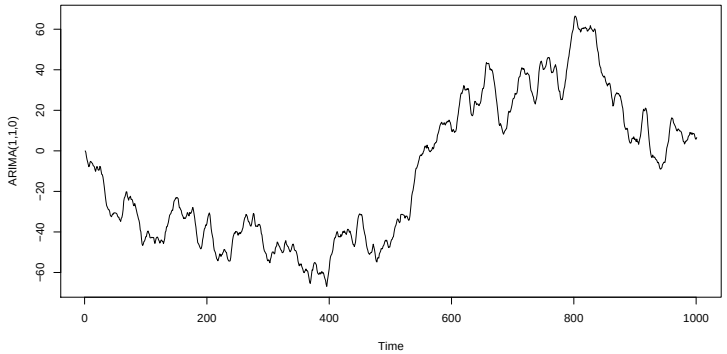



Figure 2.7: A representation from a ARIMA(1,1,0) model $0.7(B)(1 - B)y_t = 0$.

**AutoRegressive Fractionally Integrated Moving Average** We now consider a generalization of the ARIMA$(p, d, q)$ process by allowing the parameter $d$ to assume real values. An time series is said to be a autoregressive fractionally integrated moving average, ARFIMA, model if it satisfies the equation

$$(1 - \sum_{i=1}^{p} \phi_i B^i)(1 - B)^d x_t = (1 + \sum_{i=1}^{q} \theta_i B^i) + \epsilon_t \tag{2.12}$$

where $(1 - B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k$, the autoregressive $\Phi(z)$ and moving average $\Theta(z)$ polynomials satisfy the stationary and invertibility conditions for ARMA models. For in the range $-0.5 < d < 0.5$ the process is stationary and invertible and for $0.5 < d < 1$ is mean reverting. The parameter $d$ is said the long memory parameter since it controls the rate of decay of the autocorrelation function. When $d \neq 0$ the rate of decay is hyperbolic meaning that persistence in the autocorrelations: there is significant dependence between observations separated by long time intervals. Thus these models are useful for modeling time series with long memory, which is also known as long range dependence, Hurst effect or self-similarity [56].

In this work we will study three ARFIMA(1,0.4,0) models with $\phi_1 = \{-0.5, 0.5, 0.9\}$. Examples of time series generated by these models are represented in figures 2.8 to 2.10.

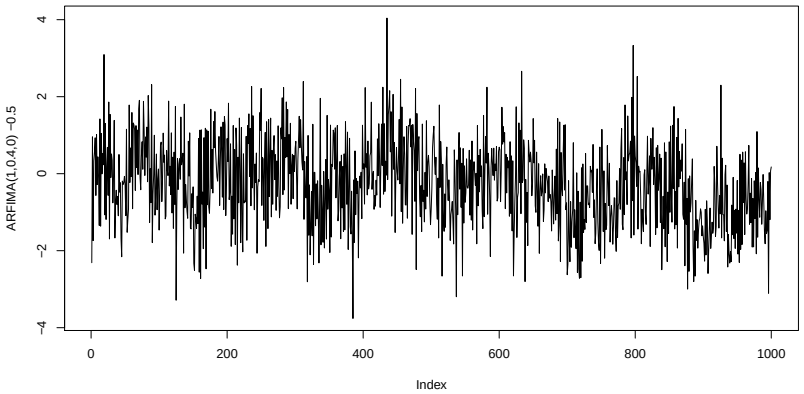



Figure 2.8: A representation from a ARFIMA(1,0.4,0) model $(1 + 0.5B)(1 - B)^{0.4} y_t = \epsilon_t$.

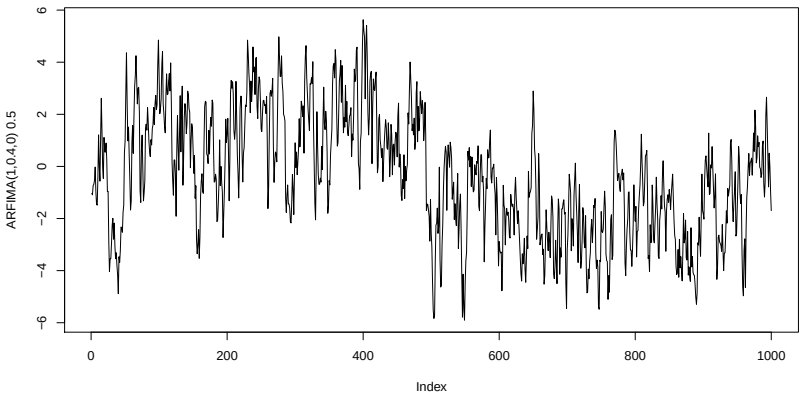



Figure 2.9: A representation from a ARFIMA(1,0.4,0) model $(1 - 0.5B)(1 - B)^{0.4} y_t = \epsilon_t$.

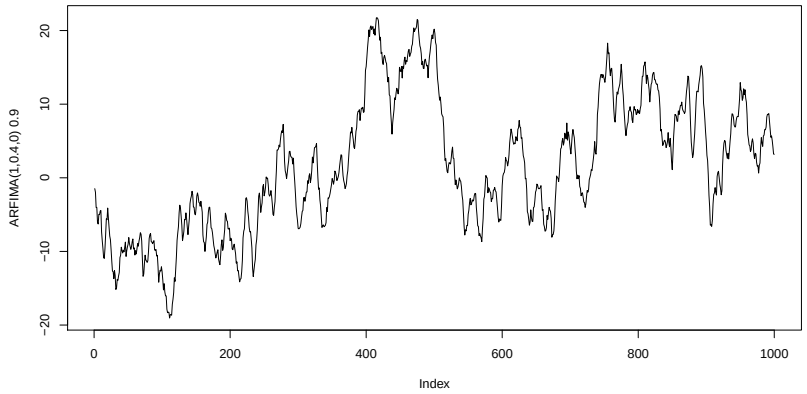



Figure 2.10: A representation from a ARFIMA(1,0.4,0) model $(1-0.9B)(1-B)^{0.4}y_t = \epsilon_t$.

#### 2.1.3.2 Nonlinear Models

Linear models have proved to be insufficient for the analysis of time series, since many of the real time series present characteristics that cannot be represented by linear models such as asymmetries of the realisations, the presence of different regimes (where the mean, variance and autocorrelation depend on the regime), sensitivity to initial conditions and volatility in clusters, that is, periods of oscillations interspersed with periods of relative calm [24]. The initial development of nonlinear time series analysis focused on several nonlinear parametric forms. We can distinguish specifications for the conditional mean and specifications for the conditional variance.

**Self-Exciting Threshold AutoRegressive** The self-exciting threshold autoregressive (SETAR) models of order 1, SETAR(1), specify the nonlinearity in the conditional mean. These are very useful for processes in which regime changes occur, where the idea is to approximate a nonlinear function in a linear function dependent on the regime that changes according to the process values [60]. This model can be presented as follows:

$$y_t = \begin{cases} \alpha y_{t-1} + \epsilon_t, & if \quad y_{t-1} \leq r \\ \beta y_{t-1} + \gamma \epsilon_t, & if \quad y_{t-1} > r \end{cases} \tag{2.13}$$

where $r$ represents a real threshold and $d$ is the delay parameter. In figure 2.11 we present an example of a representation of a time series generated by SETAR(1).

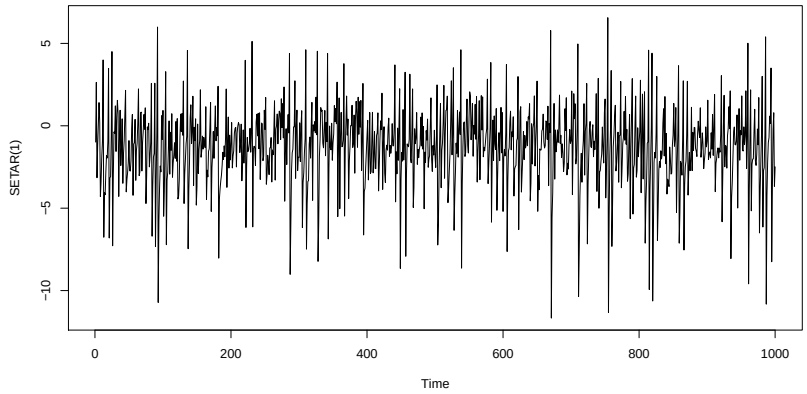



Figure 2.11: A representation from a SETAR(1) model with following parameters $\alpha = 0.5$, $\beta = -1.8$, $\gamma = 2$ e $r = -1$.

**Hidden Markov Models** The hidden Markov models (HMM) are statistical Markov models in which the system being modeled is considered a Markov process with $m$ hidden states. The model consists of two parts: firstly, an unobserved "parameter process" $C_t$ satisfying the Markov property; and secondly, the "state-dependent process" $y_t$, in which the distribution of $y_t$ depends only on the current state $C_t$ and not on previous states or observations. If the Markov chain $C_t$ has $m$ states, we call $y_t$ an $m$-state HMM [73]. This model can be presented as follows:

$$p_i(y) = Pr(y_t = y|C_t = i) \tag{2.14}$$

where $p_i$ is the probability mass function of $y_t$ if the Markov chain is in state $i$ at time $t$.

In figure 2.12 we present an example of a time series representation generated by an HMM. We chose 2 states in the hidden Markov chain and the Poisson distribution for the distribution of the observations of the process.

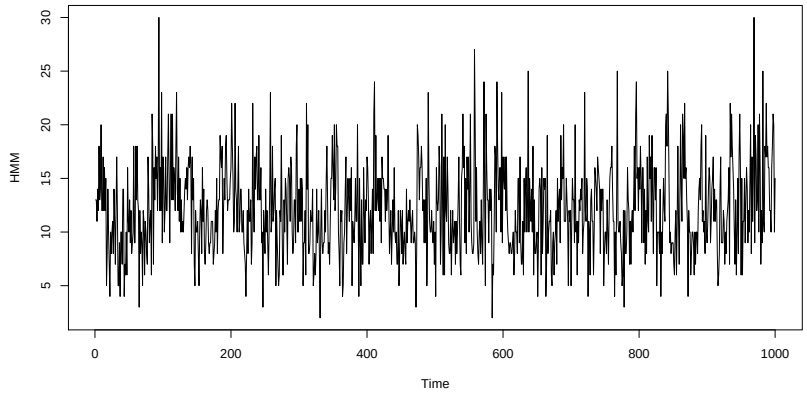



Figure 2.12: A representation from a HMM with 2 states.

**Integer-Valued AutoRegressive** The INAR models have been proposed to model correlated integer-valued time series [57]. These models are based on thinning (random) operations defined on the integers. The most common such operation is the binomial thinning defined as follows. Let $X$ be an integer valued random variable and $0 < \alpha < 1$. Then $\alpha * X = \sum_{i=1}^{X} \chi_i$ where $\chi_i \sim Be(\alpha)$, meaning that $\alpha * X|X \sim Bi(X, \alpha)$. The time series $y_t$ is said an INAR(1) if it satisfies

$$y_t = \alpha * y_{t-1} + \epsilon_t \tag{2.15}$$

where $\alpha \in [0, 1]$, $\epsilon_t$ are integer valued time series and $*$ defines the binomial thinning operation. The ACF of the INAR(1) process behaves as the ACF of an AR(1) process.

Figure 2.13 represent a realisation of a Poisson INAR(1); $\epsilon_t \sim Po(\lambda)$ that we will study in this work.

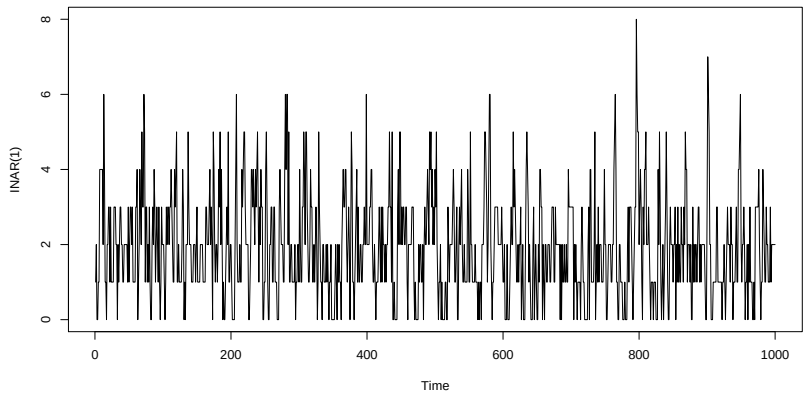



Figure 2.13: A representation from a INAR(1) model with parameter $\alpha_1 = 0.5$.

Next we consider some of the nonlinear models that are specified by the conditional variance and are widely used in finance.

In all models we will consider $\epsilon_t$ uncorrelated random variables, $z_t$ a white noise with variance 1 and $\sigma_t$ the standard deviation, so that:

$$\epsilon_t = \sigma_t z_t \tag{2.16}$$

**AutoRegressive Conditional Heteroskedasticity** An autoregressive conditional heteroscedastic (ARCH(q)) model was created to model the volatility (or conditional variance) that is not constant in time, in a homogeneous time model. The basic idea of ARCH models is that the mean-corrected asset return at is serially uncorrelated, but dependent, and that the dependence of at can be described by a simple quadratic function of its lagged values [61]. To model a time

series $\sigma_t^2$ using an ARCH process, we define

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 \tag{2.17}$$

where $\alpha_0 > 0$, $\alpha_i \geq 0$, $i > 0$ and $q$ represents the order of dependence on past shocks. The condition of non-negativity, $\alpha_i \geq 0$, ensures that the conditional variance of the error is greater than zero. An ARCH(q) process is stationary if $\sum_{i=1}^{q} \alpha_i < 1$. And the ACF of $z_t^2$ is the same as the ACF of an AR(q) process.

It is often found in economics and finance that larger values of time series also lead to greater stability (that is, greater variances), which is called (conditional) heteroscedasticity.

In figure 2.14 we present an example of a representation of a time series generated by the ARCH(2) model we will study in this work, we can verify that the values exhibit the greater variation around the peaks.

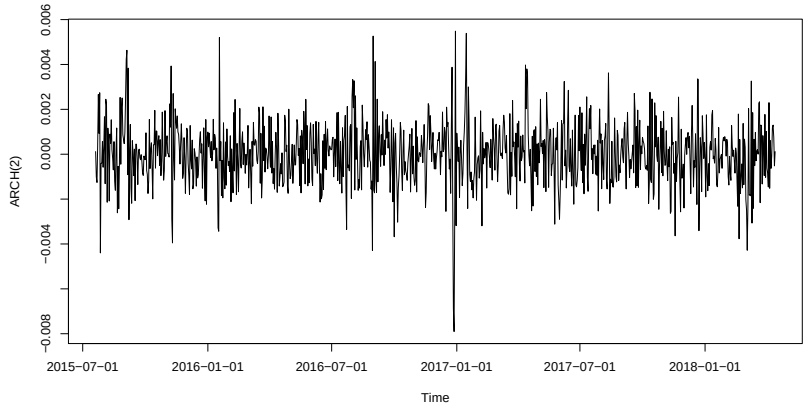



Figure 2.14: A representation from a ARCH(2) model $\sigma_t^2 = 10^{-6} + 0.4\epsilon_{t-1}^2 + 0.2\epsilon_{t-2}^2$.

**Generalized AutoRegressive Conditional Heteroskedasticity** We now consider a generalisation of the ARCH(q) process, the generalized autoregressive conditional heteroskedasticity (GARCH($p$,$q$)) processes. Propose that conditional volatility be a function not only of the squares of past errors, which is the case of ARCH models, but also of their own past values [17]. To model a time series $\sigma_t^2$ using an generalized ARCH process, we define

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 \tag{2.18}$$

where $\omega > 0$ and $\alpha_i, \beta_i \geq 0$. $p \geq 1$ represents the order of dependence of the conditional variance and $q \geq 0$ represents the order of dependence of past shocks. The conditional standard deviation can exhibit more persistent periods of high or low volatility than for an ARCH process because past values of the process are fed back into the present value.

An GARCH(p,q) process is stationary if $\sum_{i=1}^{q} \alpha_i + \sum_{i=1}^{p} \beta_i < 1$.

In this work we will study two GARCH simulated models in particular, GARCH(1,1) and GARCH(2,1), an example of a representation of each of these series can be seen in figures 2.15 and 2.16, respectively.

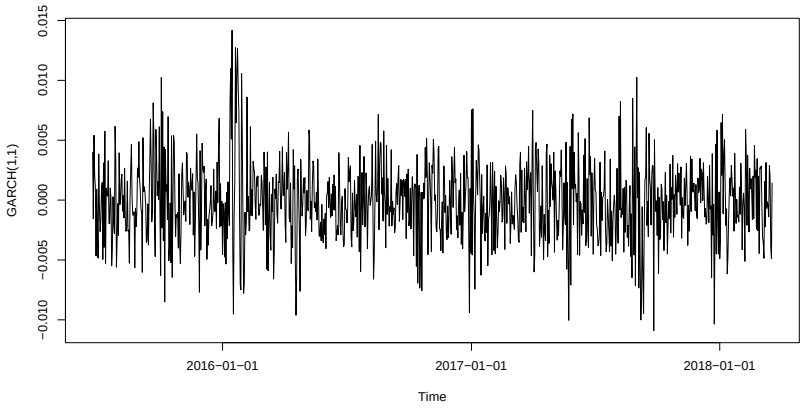



Figure 2.15: A representation from a GARCH(1,1) model $\sigma_t^2 = 10^{-6} + 0.1\epsilon_{t-1}^2 + 0.8\sigma_{t-1}^2$.

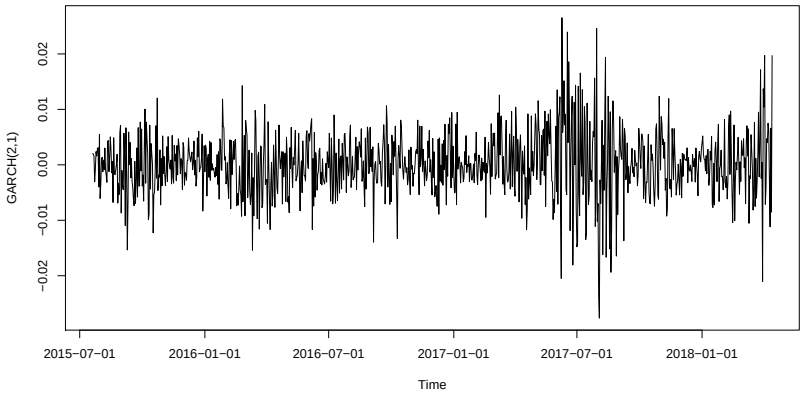



Figure 2.16: A representation from a GARCH(2,1) model $\sigma_t^2 = 10^{-6}+0.12\epsilon_{t-1}^2+0.04\epsilon_{t-1}^2+0.8\sigma_{t-1}^2$.

**Exponential Generalized AutoRegressive Conditional Heteroskedastic** Both ARCH and GARCH models treat the effect of positive and negative shocks (as asset returns in financial time series) on volatility as symmetric, which may not always be considered because innovations can cause asymmetric impacts on the volatility of a financial asset, for example. Given this vulnerability, an extension of the GARCH model was proposed, exponential GARCH (EGARCH($p$,$q$)), which assumes that this effect is asymmetric [61]. This model is given by:

$$log(\sigma_t^2) = \omega + \sum_{i=1}^{q} \alpha_i |\frac{\epsilon_{t-i}}{\sigma_{t-i}}| + \sum_{i=1}^{p} \beta_i log(\sigma_{t-i}^2) + \sum_{i=1}^{q} \gamma_i \frac{\epsilon_{t-i}}{\sigma_{t-i}} \tag{2.19}$$

where $\omega = \alpha_0 - \alpha_1\sqrt{\frac{2}{\pi}}$. The $\alpha_i$ parameters characterize the volatility grouping phenomena, $\beta_i$ is the persistence parameter, and the $\gamma_i$ parameter describes the leverage effect. This model

uses logged conditional variance to relax the positivity constraint of the model coefficients, and allows its parameters not to be restricted to be non-negative. The impact of positive shocks, $\epsilon_{t-i} > 0$ is $(\alpha_1 + \gamma_i)\frac{\epsilon_{t-i}}{\sigma_{t-i}}$, while for negative shocks is $(\alpha_1 - \gamma_i)\frac{\epsilon_{t-i}}{\sigma_{t-i}}$.

In figure 2.17 we present an example of a representation of a time series generated by EGARCH(1,1) that we will study in this work.

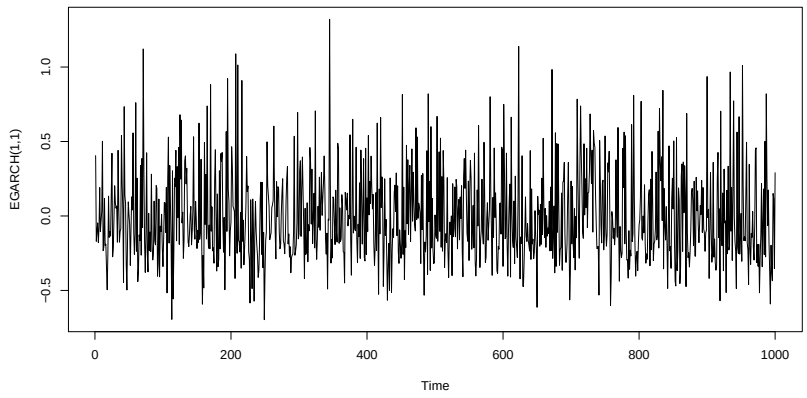



Figure 2.17: A representation from a EGARCH(1,1) model $log(\sigma_t^2) = 10^{-6} - 0.1\sqrt{\frac{2}{\pi}} + 0.1|\frac{\epsilon_{t-1}}{\sigma_{t-1}}| + 0.01log(\sigma_{t-1}^2) + 0.3\frac{\epsilon_{t-1}}{\sigma_{t-1}}$.

**Glosten-Jagannathan-Runkle Generalized AutoRegressive Conditional Heteroskedasticity** L. Glosten, R. Jagannathan, and D. Runkle proposed the Glosten Jagannathan and Runkle version of the GARCH model, denoted by GJR-GARCH($p$, $q$), as an alternative method to the use of the logged conditional variance of the EGARCH model. This model can be represented as follows:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \sum_{t=1}^{q} \gamma_i \epsilon_{t-1}^2 I_{t-i} \tag{2.20}$$

where $\omega > 0$, $\alpha$, $\beta \geq 0$, $\alpha_i + \gamma_i \geq 0$, $\beta_i + \alpha_i + \frac{\gamma_i}{2} < 1$ and the indicator function $I_{t-i} = 1$ if $\epsilon_{t-i} < 0$ and 0 otherwise.

The parameter $\gamma_i$ describes the leverage effect and the persistence parameter for this model is $\beta_i + \alpha_i\frac{\gamma}{2}$. A positive shock $\epsilon_{t-i}$ contributes $\beta_i\epsilon_{t-i}^2$ to $\sigma_t^2$, whereas a negative shock $\epsilon_{t-i}$ has a greater impact $(\alpha_i + \gamma_i)\epsilon_{t-i}^2$ with $\gamma_i > 0$. If the leverage coefficient is zero, the GJR-GARCH model will be reduced to the GARCH model.

In figure 2.18 we present an example of a representation of a time series generated by GJR-GARCH(1,1) that we will study in this work.

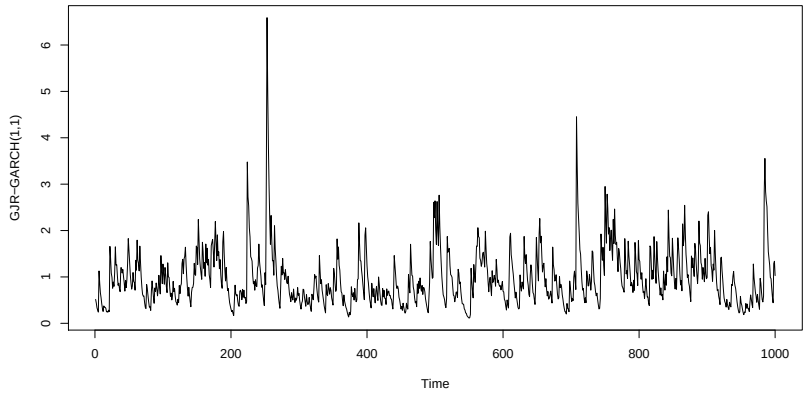



Figure 2.18: A representation from a GJR-GARCH(1,1) model $\sigma_t^2 = 10^{-6} + 0.1\epsilon_{t-1}^2 + 0.76\sigma_{t-1}^2 + 0.25\epsilon_{t-1}^2 I_{t-1}$.

## 2.2 Complex Networks

Most real world systems are complex systems, since these represent a set of elements that interact and which exhibit emergent collective properties [16]. Their study requires the focus on the interactions between the elements of the system. For this, *graphs* are a very appropriate modeling tool, in which the elements and their relations are represented by *nodes* (or *vertices*) and *links* (or *edges*), respectively. Such a graph has non-trivial topological properties, due to the specific characteristics of the complex system it represents [3]. This graphical representation is called a *complex network* and the branch of mathematics applied to such networks is called *graph theory*. We will use these two terms, graph and network, interchangeably.

A simple example of a complex network is the system constituted by web pages and the links among them. Analysing this network can help to understand how these pages are related, how they are grouped into different communities, and which are the most prominent or important. As well as a network of co-authorship, can be useful to understand how the various investigators are related, how they are grouped into different communities (one example is the scientific areas), and which researchers with greater influence.

For the first example, current search engines, such as Google, make extensive use of network structure in evaluating the quality and relevance of web pages. For producing search results, these sites evaluate the relevance of pages not only based on the number of links it receives, but also using subtle aspects of its position in the network. For example, a page can be viewed as more important if receives links from pages that are themselves important [48].

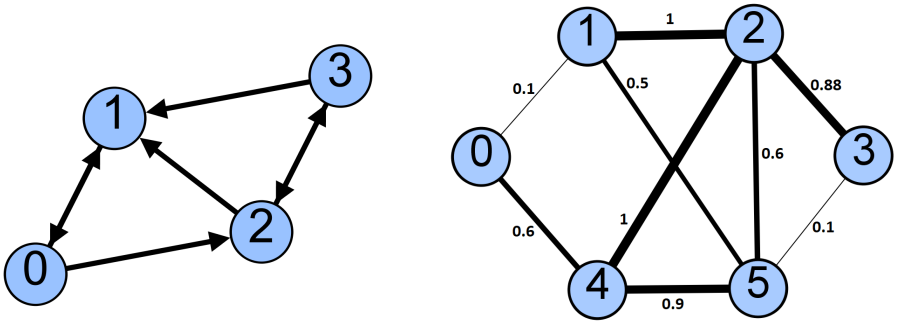
### 2.2.1 Graph Terminology and Concepts

A *graph* $(G)$ is then an ordered pair $(V(G), E(G))$, where $V(G)$ represents the set of *nodes* and $E(G)$ the set of *links* between pairs of elements of the set $V(G)$. The number of nodes, also known as the *size* of the graph, is written as $|V(G)|$ and the number of links as $|E(G)|$. A $k$-graph is a graph of size $k$.

Two nodes are *neighbors* or *adjacent* if they are connected by a link, that is, if $(v_i, v_j) \in E(G)$ then $v_i$ and $v_j$ are neighbors. We can distinguish between *directed* links, which connect a source node to a target node, and *undirected* links, when there is no such concept of orientation. In the first case the graph is called directed or *digraph*.

A graph can also be *weighted*, this means that at each link $(v_i, v_j)$ is associated with a weight (or cost) $w_{i,j}$, and this weight can be positive or negative.

A graph is classified as *simple* if it does not contain multiple links (two or more links connecting the same pair of nodes) and it does not contain self-loops (an link connecting a node to itself). Figure 2.19 illustrates a small example of a simple directed graph and a simple undirected weighted graph, respectively.




(a) Directed graph (b) Undirected weighted graph

Figure 2.19: A representation of (a) a simple directed graph and (b) a simple undirected weighted graph.

The *adjacency matrix* of a graph $G$ is denoted as $G_{Adj}$, and $G_{Adj}[i][j]$ is 1 (or $w_{i,j}$) when $(v_i, v_j) \in E(G)$ and is 0 otherwise. In the context of this thesis, all nodes are labeled consecutive integer number starting from 0 to $|V(G)| - 1$. The adjacency matrices to the graphs shown in figure 2.19 are shown in figure 2.20.

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 |

(a) Adjacency matrix

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0.1 | 0 | 0 | 0.6 | 0 |
| 1 | 0.1 | 0 | 1 | 0 | 0 | 0.5 |
| 2 | 0 | 1 | 0 | 0.88 | 1 | 0 |
| 3 | 0 | 0 | 0.88 | 0 | 0 | 0.1 |
| 4 | 0.6 | 0 | 1 | 0 | 0 | 0.9 |
| 5 | 0 | 0.5 | 0 | 0.1 | 0.9 | 0 |

(b) Adjacency matrix

Figure 2.20: Illustration of the adjacency matrices of (a) a simple directed graph and (b) a simple undirected weighted graph.

**Path** A *path* is a sequence of nodes in which each consecutive pair of nodes in the sequence is connected by an link.

**Connectivity** The concept of *connectivity* is extremely important in networks. We say that two nodes are then *connected* if there is a path between them and are *disconnected* if such a path does not exist. In undirected graphs, if nodes $v_i$ and $v_j$ are connected and nodes $v_j$ and $v_k$ are connected, then $v_i$ and $v_k$ are also connected. This property can be used to partition the nodes of a graph in non-overlapping subsets of connected nodes known as *connected components* [15]. Although there is at least one path between any two nodes in the same component, there is no path between nodes belonging to different components [6]

**Subgraph** A subgraph $G_k$ of size $k$ of a graph $G$ is a graph with $k$ nodes in which $V(G_k) \subseteq V(G)$ and $E(G_k) \subseteq E(G)$.

### 2.2.2 Topological Metrics

There is a vast set of topological metrics of available graphs [15], each reflecting some particular features of the system under analysis. The majority of them can be distinguished between local and global measures, that is, those related to individual nodes or links, and those related to the graph as a whole. In our work we are essentially concerned with studying global metrics. We present below a detailed description of them as well as some other local measures for a better understanding.

**Average Degree** The degree of a graph is a fairly important local property of each node, this represents the number of links that the node has for the other nodes, in undirected graphs [6].

We denote by $k_i$ the degree of the $i$-th node.

In digraphs, we distinguish between the *in-degree*, $k_i^{in}$, and the *out-degree*, $k_i^{out}$. The first represents the number of links that point to node $v_i$, and the second the number of links that point from node $v_i$ to other nodes. The total degree, $k_i$, in a digraph is given by:

$$k_i = k_i^{in} + k_i^{out} \tag{2.21}$$

In many networks, there are some nodes that have a fairly high degree compared to others. We call these nodes *hubs*.

In weighted graphs we may want to obtain the weighted degree that is similar to the previous measure (undirected and directed). The difference is that here we take into account the weight of the connections, that is, instead of adding the quantity of connections, we sum the weights of each of the links [15]:

$$k_i^w = \sum_{j=1}^{|V(G)|} w_{i,j} \tag{2.22}$$

From this measure we can obtain the average degree ($\bar{k}$) which is a fairly important global measure and can be easily obtained by calculating the arithmetic mean of the degrees of all nodes in the graph, mathematically [6]:

$$\bar{k} = \frac{1}{|V(G)|} \sum_{i=1}^{|V(G)|} k_i \tag{2.23}$$

**Average Path Length** A path is a sequence of nodes in which each consecutive pair of nodes in the sequence is connected by a link. It may also be useful to think of the path as the sequence of links that connect those nodes. In digraphs the path follows the direction of the source node for the target node. Paths can contain repeated nodes or not, the paths without repeated nodes are called simple paths. Each path consists of $n + 1$ nodes and $n$ edges.

We denote by $\bar{d}$, the arithmetic mean of the shortest paths ($d$) among all pairs of nodes (both ways for directed graphs), the path length being the number of links, or the sum of the links weights if the graph is weighted, in the path [6]. It should be noted that $\bar{d}$ is measured only for the node pairs that are in the same component. The function we use in our thesis to calculate this metric does not consider link weights currently and uses a breadth-first search.

It is a good measure of the efficiency of information flow on a network.

For a directed graph the average path length is given by:

$$\bar{d} = \frac{1}{|V(G)|(|V(G)|-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{|V(G)|} d_{i,j} \tag{2.24}$$

**Global Clustering Coefficient** This measure ($C$), also called global transitivity, measures the total number of closed triangles in the graph, that is, it measures the degree to which the nodes in a graph tend to cluster. It is calculated by the ratio of the number of closed triangles ($N_\triangle$) to the number of possible triangles, that is, the amount of connected triplets of nodes ($N_3$) [15]. In this work we will call it only as a *clustering coefficient*.

For undirected and unweighted graphs, we mathematically have to:

$$C = \frac{3N_\triangle}{N_3} \tag{2.25}$$

Factor three explains the fact that each triangle can be seen to consist of three different triangles, one with each of the nodes as the central node, and ensures that $0 \leq C \leq 1$. For directed graph the direction of the edges is ignored.

For weighted graphs there are several generalizations of clustering coefficient, here we use the definition by A. Barrat [7], this is a local vertex-level quantity, its formula is:

$$C_i = \frac{1}{k_i^w(k_i - 1)} \sum_{j,h} \frac{(w_{i,j} + w_{i,h})}{2} a_{i,j} a_{i,h} a_{j,h} \tag{2.26}$$

where $k_i^w$ is the weighted degree, $a_{i,j}$ are elements of the adjacency matrix, $k_i$ is the degree and $w_{i,j}$ are the weights.

**Number of Communities** Measures the number of denser subgraphs in a network, that is, subsets of nodes within the graph such that connections between the nodes are denser than connections to the rest of the network.

The function we use in our thesis to help us calculate this metric tries to find densely connected subgraphs, also called communities here, via random walks. The idea is that short random walks tend to stay in the same community. This function is the implementation of the Walktrap community finding algorithm [50].

In our work we denote as $S$ the number of communities.

**Modularity** Measures how good the division of the graph is in specific communities, that is, how different are the different nodes, belonging to different communities, from each other. A high modularity value, ($Q$), indicates a graph with a dense internal community structure, that is, with many edges between nodes within communities and sparse connections between nodes of different communities [14].

If a particular network is split into $c$ communities, $Q$ can be calculated from the symmetric $c \times c$ mixing matrix $E(G)$ whose elements along the main diagonal, $e_{ii}$, give the fraction of connections between nodes in the same community $i$ while the other elements, $e_{ij}(i \neq j)$ identify the fraction of connections between nodes in the different communities $i$ and $j$. The calculation of $Q$ can then be performed as follows [14, 15]:

$$Q = \sum_i [e_{ii} - (\sum_j e_{ij})^2] \tag{2.27}$$

The situation $Q = 1$ identifies networks formed by disconnected modules.

### 2.2.3 Basic Complex Networks Types

Given these statistical properties, complex networks can be classified as regular networks, random networks, scale-free networks, small-world networks, among other. Below we briefly describe the types mentioned.

**Regular Networks** *Regular networks* are graphs whose nodes have all the same degree, that is, each node has the same number of neighbors. In physics, for example, atomic models are studied through regular networks.

**Random Networks** *Random networks* are graphs obtained by a random process, that is, they can be variables as a sample of a probability distribution, since in this model, all nodes have more or less the same number of connections, or the same probability of receiving new ones connections. Random networks refer almost exclusively to the Erdös-Rényi random graph model.

Erdös and Rényi represented the formation of social networks, and for them, a connection between each one of the guests of a party would be enough, so that all were connected at the end of it. According to them, the more links were added, the greater the probability of being generated clusters, that is, groups of more cohesive nodes.

**Scale-Free Networks** *Scale-free networks* are complex networks whose degree of distribution follows the power law, (a functional relation between two quantities, the change of one quantity results in a proportional change in the other quantity), in which most nodes have few links in contrast to the existence of some nodes with a high degree, that is, nodes with high degree tend to connect to nodes with also high degree.

The oldest published example of a scale-free network is probably the network of citations between Price's scientific works [46], another example is web page networks.

**Small-World Networks** The small-world phenomenon resulted from experiments carried out by Stanley Milgram in the 1960s The results in which letters passed from person to person were able to reach a target individual in only a small number of steps, around 6 steps were one of the first manifestations of the effect of the small-world. This phenomenon consists in the fact that most of the pairs of nodes, in most networks, seem to be connected by a short path in the network [46].

*Small-world networks* are then networks in which most nodes have few links, but can be reached from another node through a small number of links, that is, the shortest path between any pair of nodes grows proportionally to the logarithm number of nodes in the network. One example is the social networks.

# Chapter 3

# From Time Series to Complex Networks

Univariate time series analysis is a mature and solid field with well developed and understood methods and associated theory. There is also a plethora of models to describe the behavior of particular series, ranging from the linear ARMA models for stationary time series and S(easonal)ARIMA models particularly suited to forecasting macroeconomic data to nonlinear GARCH type models for modelling conditional volatility and widely applied in financial data [34, 56]. However, time series analysis still has some limitations when it comes to the study of more complex signals. In particular, the relevant effect of nonlinearities on their mathematical representations, which is a characteristic that can be reflected in the temporal evolution of the variable (or variables) that describes the system and requires the use of specific tools for its analysis. Furthermore, the study of dynamic systems such as long-range correlated stochastic processes, chaos, intermittency, among other complex systems lead to the analysis of time series for the limits of their methods.

This type of analysis is used in a variety of important domains, such as climate studies [45], health monitoring [33] and financial data analysis [61]. Recently, with the technological advances such as the rapid development and adoption of the Internet of Things concept, there has been an tremendous increase in the availability of time series data generated from heterogeneous sensors and wearable. For example, sensors can measure pollutants and various meteorological variables across regions, parameters of a medical patient, or fitness parameters of a healthy person. Other examples where massive time series data are generated include web applications tracking user clicks or point-of-sales data in a large store.

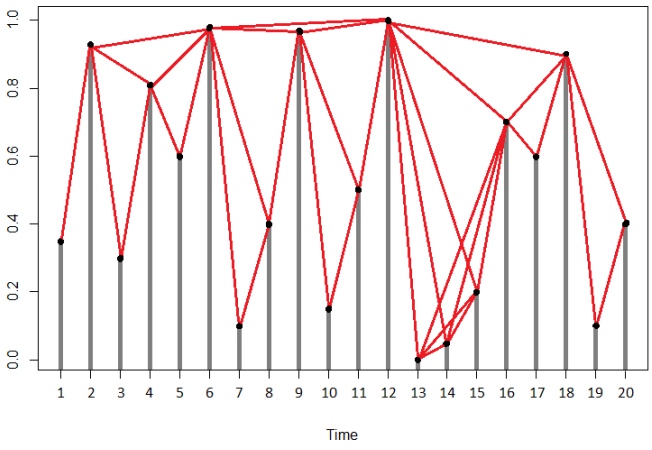
Due to the large quantities and diversity of time series and the imminent interest in extracting useful knowledge from such data, new approaches to dealing with traditional analysis limitations of such series are being developed. Namely, motivated by the success of complex network methodologies, several network-based time series analysis approaches have been recently proposed, based on mapping time series to the network domain. The mappings proposed in the literature

are based on concepts such as correlation [70], phase space reconstruction [26], recurrence analysis [43], visibility [36] or transition probabilities [55].

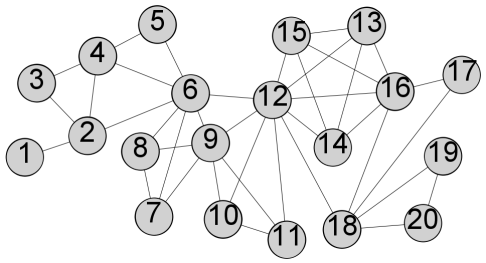
In this chapter, we perform a detailed description of the time series mapping methods in complex networks that we apply in our work, as well as some of the main results obtained in the literature. These mappings are based on concepts of visibility and transition probabilities. We also briefly present other types of existing mappings and their applicability, for a better knowledge of the state of the art.

## 3.1 Natural Visibility Graph

The first method based on the concept of visibility was first proposed in 2008 by Lacasa et al. [36]. Named natural visibility graph (NVG), each node in the graph corresponds, in the same order, to the time series data and two nodes are connected if there is a line of visibility between the corresponding data points, that is, if it is possible to draw a straight line in the time series that joins the two corresponding data points that intercepts no data "height" between them. This method is based on the idea that each observation of the time series is seen as a vertical bar with height equal to the corresponding numerical value and that when we consider these vertical bars in a landscape, the upper part of a bar is visible by the upper part of other bars. If we consider each time instant as a node of a graph, then two nodes are connected if the tops of the corresponding vertical bars are visible to each other, that is, if there is a straight line from the top of the two bars that does not intersect other bars. This mapping is illustrated in figure 3.1 with a toy time series and the resulting network.




(a) Toy time series




(b) NVG

Figure 3.1: On the left side, we present the plot of a toy time series and, on the right side, the network generated by the natural visibility algorithm. The red lines in the time series plot represent the lines of visibility (and hence the links of the graph) between all data points.

The resulting graph has as many nodes as the number of observations in the time series. The nodes are numbered sequentially in time and each node corresponds to an observation $(t_a, y_a)$. Two nodes $(t_a, y_a)$ and $(t_b, y_b)$ are connected (have visibility) if any other observation $(t_c, y_c)$ with $t_a < t_c < t_b$ satisfies:

$$y_c < y_b + (y_a - y_b)\frac{(t_b - t_c)}{(t_b - t_a)} \tag{3.1}$$

The graphs obtained always have the following characteristics [36]:

- **Connected:** each node sees at least its nearest neighbors (left-hand side and right-hand side).
- **Undirected:** the way the algorithm is built up, there is no direction defined in the links. However, this direction could be defined considering the direction of the time axis.
- **Invariant under affine transformations of the series data:** the visibility criterion is invariant under rescheduling of both the horizontal and vertical axis, as well as in horizontal and vertical translations.
- **"Lossy":** some information regarding the time series is inevitably lost in the mapping from the fact that the network structure is completely determined in the (binary) adjacency matrix. For instance, two periodic series with the same period as $Y_a = ..., 3, 1, 3, 1, ...$ and $Y_b = ..., 3, 2, 3, 2, ...$ would have the same visibility graph, albeit being quantitatively different. One possible solution would be the use of weighted networks, where weights determine the height difference of the associated data, for example.

An NVG typically has a distinct topology characterized by hubs representing the local maximums of the corresponding time series. The presence of these hubs gives rise to a community structure, where the different communities reflect the temporal order of observations [20].

A disadvantage of a direct application of this method is that it is not suitable to distinguish nonstationary time series with increasing trend of series with decreasing trend. For example, if we consider any time series with increasing trend and a series that is symmetric to that (decreasing trend), the graphs obtained will be identical, and consequently, all their properties. But this disadvantage can be overcomed if we use directed links.

The main advantage of this method is that it is easy to implement. Furthermore, like it is the case for all mappings here presented, there is a vast arsenal of well-defined methods and techniques of complex networks that can later be used to study the behavior and structure of the resulting networks.

In terms of computational complexity, a naive generation of NVGs without optimization has quadratic temporal complexity ($\mathcal{O}(n^2)$, where $n$ is the length of the time series), which can be

a prohibitive cost for very long time series. However, a more efficient algorithm based on the divide and conquer technique was proposed in [38], with a temporal complexity of $\mathcal{O}(n \log n)$.

**Main Results in the Literature** Lacasa et al. [36] showed that NVGs inherit several structural properties of time series, since the method maps periodic series on regular graphs, random series on random graphs and fractal time series on scale-free graphs. More specifically, in the case of periodic series, the resulting graphs represent a motif concatenation, that is, a repetition of a pattern associated with the time series period. In the case of the fractal time series, it is shown that the degree distributions of power law will be related to fractality.

Later, Lacasa et al. [37], studied in more detail the fractional Brownian motion that can be characterized by the so-called Hurst exponent ($0 < H < 1$), which is widely used to study real time series, but requires sophisticated techniques due to its nonstationarity and dependence long range. The authors verified that the degree distribution of the resulting scale-free visibility graph is a Hurst function, insofar as the power law distribution exponent depends linearly on the Hurst value, which is an alternative and easy method to calculate the Hurst exponent. They have shown that this result is also valid in the case of fractional Gaussian noises and generic noises $f^{-\beta}$.

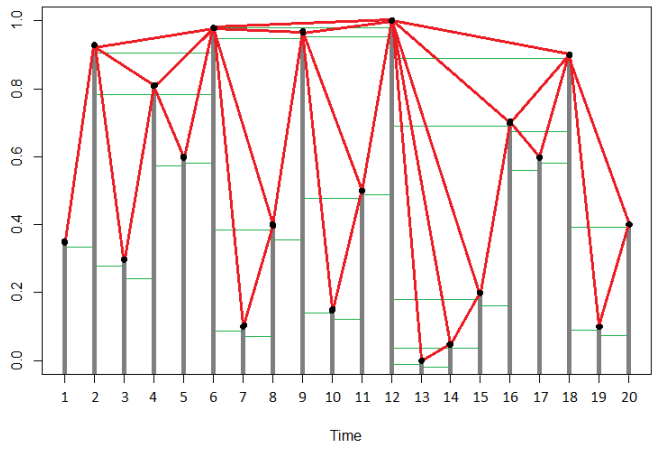
Given the potential demonstrated it was believed that the method could be applied to study problems such as turbulence, stock market dynamics, physiological signs and others. And quickly began to emerge the first work in this direction, in particular, [40] in the study of energy dissipation rates in three-dimensional turbulence, [51, 69, 72] in the study of financial series, and [54] in study of heart rate series.

We quickly began to think of variants of these graphs in order to explore different characteristics. One of the most important and most explored variants is called horizontal visibility graphs (HVG), which we will describe in more detail below. Another version is a parametric NVG (PNVG) [8], where NVG links must satisfy a constraint determined by a parameter, the angle of view. This restriction translates into a change in the properties of graphs, such as average degree, average path length and number of communities obtained. Another variant, weighted visibility graph (WVG) [58], is the addition of weights and direction to NVG links. These weights are given by the tangent arc, angle of view, between two points in the time series and are aimed at detecting sudden fluctuations in electroencephalogram (EEG) signals of an epileptic convulsive activity. Another variant called limited penetrable visibility graph (LPVG) [25] was proposed to present high anti-noise capacity of the time series, especially real time series that inevitably have noise.

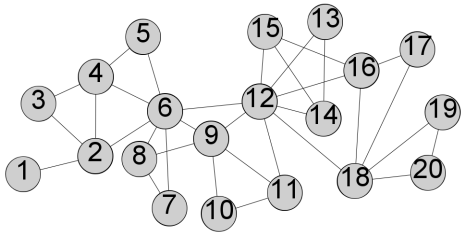
## 3.2 Horizontal Visibility Graph

In order to reduce the computational complexity associated to NVG, Luque et al. [42] proposed in 2009 a simplified NVG method called the horizontal visibility graph (HVG), which inherits all NVG features mentioned above.

In this alternative, two nodes in the graph are connected if it is possible to draw a horizontal line in the time series joining the two vertical bars, corresponding to the two data, which does not intercept any height of the intermediate data. In the figure 3.2 we give a simple illustration of this method, with a toy time series and the resulting network.




(a) Toy time series




(b) HVG

Figure 3.2: On the left side, we present the plot of a toy time series and, on the right side, the network generated by the horizontal visibility algorithm. The green lines represent the horizontal lines of visibility between all the data points and the red lines the respective connections between the points.

Formally, two nodes $(t_a, y_a)$ and $(t_b, y_b)$ are connected, have visibility, if the following condition is fulfilled:

$$y_a, y_b > y_c \tag{3.2}$$

for all $t_c$ such that $t_a < t_c < t_b$.

The HVG is always a subgraph of the NVG associated with the same time series, for example, if we analyze both graphs in the figures 3.1 and 3.2 we can easily verify that all the links present in the HVG are present in NVG, but there are links in NVG that are not in HVG, two examples are links $(7, 9)$ and $(13, 15)$. Therefore, the HVG nodes will always have a degree less than or equal to the nodes of the corresponding NVG, since they will have "less visibility" and consequently will have less quantitative information. Although this fact has not impacted on the

qualitative characteristics of the graphs, in a quantitative way, the HVGs will typically have "less statistics" [42] than the NVGs, which for the analysis of certain time series may be crucial.

In terms of computational complexity, the generation of HVGs if really efficient and can be almost trivially obtained with linear temporal complexity ($\mathcal{O}(n)$, where $n$ is the length of the time series) [42].

**Main Results in the Literature** Luque et al. [42] have formally shown that HVG is a fairly simple method to discriminate randomness in time series, since any uncorrelated random time series is converted into a graph where:

- the degree distribution is exponential and has the following form: $p_k = (\frac{1}{3})(\frac{2}{3})^{(k-2)}$ to $k \geq 2$,
- the local clustering coefficient is: $C_i(k) = \frac{2}{k}$, and
- the probability that two data separated by $n$ intermediate data are two nodes connected in the graph can be demonstrated as: $P(n) = \frac{2}{n(n+1)}$,

since the HVG is a small-world random graph.

The authors also studied the possibility of a functional relationship between the value (bar height) of an observation and the degree of its corresponding node, where they concluded that the hubs correspond to the observations with the highest value (higher bars), that is, the extreme values of the series. Another result indicates that the average path length increases with increasing the size of the graph.

In [35] the authors were able to use the HVGs to characterize and distinguish time series generated by correlated, uncorrelated and chaotic processes. Although all of these series are mapped to graphs with exponential degree distribution of the form $p_k \sim exp^{(-\lambda k)}$, they showed that the value of $\lambda$ can characterize the different processes. Specifically, if $\lambda < \frac{3}{2}$ the process is chaotic, if $\lambda > \frac{3}{2}$ the process is correlated, and if $\lambda = \frac{3}{2}$ the process is uncorrelated.

A recent work [39] shows that the topological properties (such as small probability distribution of motifs, density, assortativity, etc...) of NVGs and HVGs are useful features for constructing accurate classification models using generic classifiers for the classification of real time series, since the junction of both graphs allows you to capture global (in the case of NVGs) and local (in the case of HVGs because they are more sensitive to local variations) features of a variety of time series. They concluded that such a junction allows to increase the classification accuracy of a vast set of time series of different domains and that this new network approach may in fact be an efficient approach to time series classification problems when compared to the commonly used approaches.

In [30] they extracted motifs from HVGs and showed that the motifs statistic can be used to differentiate several types of time series, for example white Gaussian process and fully chaotic processes with noise. They also showed that the motif profiles of HVGs generated from time series of heart rate can group different types of meditative activities.

A variant of the NVG and HVG consists of the creation of difference visibility graphs (DVG), which consists of subtracting the set of edges from the HVG to the set of edges of the NVG in order to extract sequences of essential degrees, this variant was used for automatic classification of the stages of sleep [71]. Similar to the LPVG version of NVG, [25] also proposed the multispanned limited penetrable horizontal visibility graph (MLPHVG), to present high anti-noise capability of the time series.

HVGs were also used to analyze seismic signals [59] and to evaluate the complex dynamics of tourism systems [5].

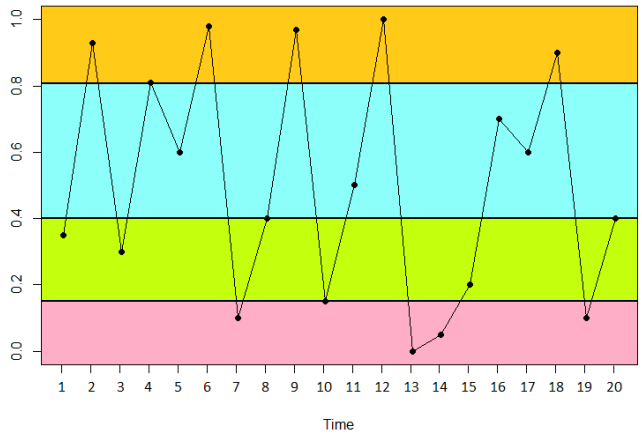
## 3.3 Quantile Graph

Quantil graphs (QG) were introduced by Campanharo et al. [11]. It is a different approach from previous methods of visibility, but captures pertinent information from time series, namely oscillations over time. This method divides the time series into $Q$ quantiles, $q_1, q_2, ..., q_Q$, and each quantile, $q_i$, is associated to a node $v_i$ of the graph. So the graph has as many nodes as the number of quantiles. Two nodes $v_a$ and $v_b$ are connected by a weighted directed link $(v_a, v_b, w_{a,b})$, where the weight $w_{a,b}$ represents the number of times an observation $(t_n, y_n)$, belonging to the quantile $q_a$, is followed by an observation $(t_{n+1}, y_{n+1}$, belonging to the quantile $q_b$. The weights are normalized such that the adjacency matrix becomes a Markov transition matrix, where $\sum w_{a,b} = 1$. This mapping is illustrated in figure 3.3 with a toy time series and the resulting network.

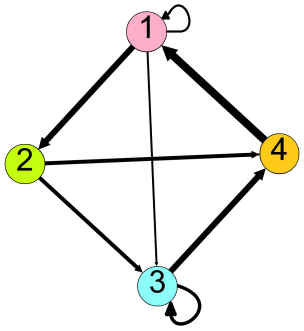
The number of quantiles is usually much less than the length of the time series and the resulting networks are weighted and directed. If the number of quantiles is too large for the range of values that the time series represents, the resulting graph will not be connected, that is, it may contain node sets that will not be connected to any other node. This happens because the series will be divided into so many quantiles that some of them will not contain any of the observations of the series.

These networks have a significant loss of information on small amplitude variations, especially if the value of $Q$ is very small. Its connectivity represents the causal relationships contained in the dynamics of the process it represents.

To refer to a QG of $q$-nodes we use the following nomenclature: $q$-QG or simply Q$q$.




(a) Toy time series




(b) QG

Figure 3.3: On the left side, we present the plot of a toy time series and, on the right side, the network generated by the quantiles algorithm. The different colors represent the region (in the time series plot) corresponding to the different quantiles (in this case $Q = 4$). Repeated transitions between quantiles result in edges in the network with larger weights represented by thicker lines.

**Main Results in the Literature** Campanharo et al. [11] used this method to propose an inverse mapping, that is, a mapping that converts a complex network into the corresponding time series without any *a priori* knowledge of the correspondence between nodes and quantiles, so that there is the possibility of studying the structure and the properties of the networks using time series analysis.

The authors were able to show that time series with different properties are mapped in networks with different topological properties. In particular, as time series become more random, corresponding networks become increasingly random. They have also shown that the method is useful for finding subtle differences between time series of heart beats in healthy and unhealthy peoples. They also showed that the results were maintained against different values of chosen quantiles as well as for series of different healthy people and different unhealthy people.

In [19], they have shown that QGs are capable of capturing and quantifying time series derived from long-range correlated processes and chaotic processes. They also verify that as the time series changes from periodic to chaotic, the corresponding networks, initially regular, become more and more random with larger values of average path length and clustering coefficient.

In another work [10], the authors applied the method to time series of EEG for the detection of epileptic behavior, and obtained networks with clear differences in their topology, namely, the QG corresponding to the healthy people signal presents a voluminous topology due to to the pattern of variability found in the corresponding time series, (jumps much between quantiles), whereas the QG corresponding to the signal of the people with epilepsy presents an elongated chain pattern due to the peaks present in the corresponding time series. These characteristics are reflected in the clustering coefficient and average path length, which are higher for epilepsy case.

## 3.4 Other Mappings

In addition to the mapping methods presented there are other variations as well as methods based on concepts different from those presented. One of the most mentioned and explored is the correlation-based method, Zhang and Small [70] were the first to build a complex network from a pseudo-periodic time series, based on this method, and capture the linear characteristics of the series. Here nodes can represent cycles (in the case of Zhang and Small), phase space or other characteristics, whereas the links are determined by the force of correlation, usually defined by an empirical threshold. The authors verified that periodic time series with noise are mapped into random networks, while time series from chaotic processes resulted in scale-free and small-world networks.

Another method is based on the incorporation of the time series into an appropriate phase space, where we represent each point of phase space on a node of the network and select a fixed number ($k$) of neighbors closer to each point, and connect each node with its $k$ nearest neighbors. Using this method, Xu, Zhang and Small [68], found different superfamilies for different time series (periodic, periodic with noise, chaotic, hyper chaotic, white noise and fractal noise), based on the relative frequency of motifs found in the corresponding networks.

Recurrence networks were also proposed to analyze the structural properties of time series, which is based on the recurrence matrix of their observations, since recurrence is a fundamental property of many dynamic processes. In [20], using the recurrence matrix obtained and a threshold on the minimum distance between two points (as in [68]) in the phase space as the adjacency matrix of an undirected and unweighted graph, of systems widely studied (as of Hénon map, Bernoulli map, Rössler system and Lorenz system). Here they came to a correspondence between some complex network metrics and phase space properties, concluding that recurrence networks preserve all topologically relevant phase space information of the system.

# Chapter 4

# Characterization of Time Series using Topological Features

Statistical time series analysis relies on a set of concepts, measures and models designed to capture the essential features of the data, namely autocorrelation, trend, cyclical fluctuations, heteroscedasticity. However, the available methodologies present severe limitations for many time series datasets, for example for multidimensional time series, very short or very long time series, irregularly spaced observations, to name just a few. Complex networks based analysis of time series relies on several mappings proposed in the literature to capture specific characteristics of the data (c.f chapter 3).

The aim of this chapter is to characterize the time series models that are most common and widely used in the statistical theory and practice of time series analysis in terms of the topological features obtained from mapping the time series processes to complex networks. To this end we perform a detailed simulation study. Specifically, we generate 100 sample paths of size $T$ of each of the models presented in section 4.1, in a total of 1700 time series. The time series are then mapped into networks using the NVG, HVG, 10-QG, 50-QG and 100-QG methods presented in chapter 3. The resulting 8500 ($1700 * 5$) networks are characterized by the global topological metrics mentioned in subsection 2.2.2: average degree, average path length, number of communities, clustering coefficient, and modularity.

Since the range of values of the metrics present a wide variation across the models, we consider the *Min-Max* normalization defined as follows.

Let $X = (x_1, x_2, ..., x_n)$ be a vector of values (of a topological measure). The Min-Max normalized vector $X^{'} = (x_1^{'}, x_2^{'}, ..., x_n^{'})$ satisfies

$$x_i^{'} = \frac{x_i - \min_{1 \leq j \leq n} x_j}{\max_{1 \leq j \leq n} x_j - \min_{1 \leq j \leq n} x_j} \tag{4.1}$$

yielding $0 \leq x_i^{'} \leq 1$.

The normalized topological measures for each model, organized by mapping, are summarized in tables by the mean and standard deviation and represented in boxplots.

Section 4.1 presents the results obtained from simulating the models described in section 2.1.3. Furthermore, section 4.2 presents the results of a similar study performed for a public time series dataset, the *Synthetic Control* dataset, available at http://archive.ics.uci.edu/ml/machine-learning-databases/synthetic_control-mld/.

All the computations are performed in `R` [52] (version 3.4.4), using using specific packages, such as `timeSeries` [66], `fracdiff` [47], `fGarch` [67], `rugarch` [27], and `HMMpa` [65], for the simulation of some of the time series models and `igraph` [18] for graph generation and calculation of metrics. The simulation of some time series models required the implementation of the appropriate procedure.

## 4.1 Time Series Models

We generate a set of 1700 time series of length $T = 10000$ corresponding to 100 replicates of each of the following 17 linear and nonlinear models:

- White noise generated by: $\epsilon_t \sim N(0, 1)$. We refer to this model as `WN`.
- Linear models

  **AR** generated by equation (2.9) with parameters $\phi_1 = \{-0.5, 0.5, 0.9\}$ for AR(1) processes and $\phi_1 = 1.5$ and $\phi_2 = -0.75$ for the AR(2) process. The choice of parameters ensures that the time series present a range of characteristics. $\phi_1 = 0.5$ leads to smoother time series than $\phi_1 = -0.5$ and $\phi_1 = 0.9$ generates time series near nonstationarity. $\phi_1 = 1.5$ and $\phi_2 = -0.75$ for the AR(2) generates pseudo-periodic time series. We refer to these four models as `AR(1)-0.5`, `AR(1)0.5`, `AR(1)0.9` and `AR(2)`, respectively.

  **ARIMA** generated by equation (2.10) with $p = 1, d = 1, q = 0$ and $\phi_1 = 0.7$. The time series generated present stochastic trend. We refer to this model `ARIMA`.

  **ARFIMA** generated by equation (2.12) with $p = 1, d = 0.4, q = 0$ and $\phi_1 = \{-0.5, 0.5, 0.9\}$. With these parameters we generate stationary time series exhibiting long memory, meaning that the autocorrelation decays slowly. We refer to these three models as `ARFIMA-0.5`, `ARFIMA0.5` and `ARFIMA0.9`, respectively.

- Nonlinear models

  **SETAR** generated by equation (2.13) with parameters $\alpha = 0.5$, $\beta = -1.8$, $\gamma = 2$ and $r = -1$. The two regimes have quite different autocorrelation properties: in the first the correlation is positive while in the second alternates between positive and negative values. We refer to this model only as `SETAR`.

**HMM** generated by equation (2.14) with 2 states in the hidden Markov chain and transition matrix: $\left[\begin{smallmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{smallmatrix}\right]$. This model generates data from a Poisson distribution with parameter 10 for the first regime and parameter 15 for the second. We refer to this model only as `HMM`.

**INAR** generated by equation (2.15) $\alpha_1 = 0.5$ and Poisson arrivals with $\lambda = 1$. This model generates integer valued data with autocorrelation decaying at a rate of 0.5. We refer to this model as `INAR`.

The next models are widely used in finance to model conditional heterocesdaticity and asymmetry and the parameters are set to *usual* values obtained from modelling financial time series.

**ARCH** generated by equation (2.17) with $q = 2$, $\alpha_0 = 10^{-6}$, $\alpha_1 = 0.4$ and $\alpha_2 = 0.2$. We refer to this model only as `ARCH`.

**GARCH** generated by equation (2.18) with $p = 1, q = 1, \omega = 10^{-6}$, $\alpha_1 = 0.1$ and $\beta_1 = 0.8$, and $p = 2, q = 1$, $\omega = 10^{-6}$, $\alpha_1 = 0.12$, $\alpha_2 = 0.04$ and $\beta_1 = 0.8$. We refer to these two models as `GARCH1` and `GARCH2`, respectively.

**EGARCH** generated by equation (2.19) with $p = 1, q = 1$, $\omega = (10^{-6} - 0.1\sqrt{\frac{2}{\pi}})$, $\alpha_1 = 0.1$, $\beta_1 = 0.01$ and $\gamma_1 = 0.3$. We refer to this model only as `EGARCH`.

**GJR-GARCH** generated by equation (2.20) with $p = 1, q = 1$, $\omega = 10^{-6}$, $\alpha_1 = 0.1$, $\beta_1 = 0.7$ and $\gamma_1 = 0.25$. We refer to this model only as `GJR-GARCH`.

The topological features obtained from the 1700 time series are summarized in tables 4.1 to 4.5, each table reporting the mean and standard deviation (in brackets) of the standardized (across models) metrics. For easiness of interpretation, the columns of the tables are colored with a gradient based on the mean values: cells with a maximum value of 1 are colored red, cells with the minimum value 0 are colored white and the remainder with a hue of red color proportional to its value.

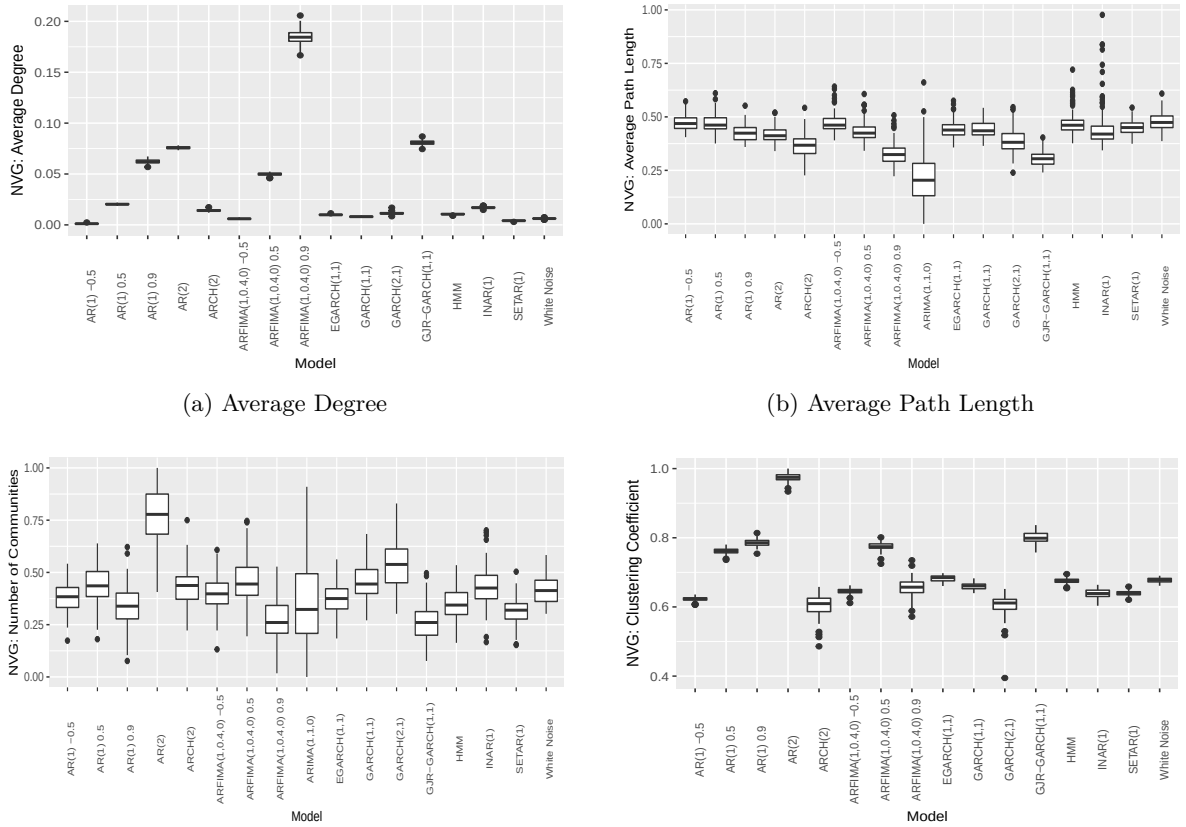
### 4.1.1 Natural Visibility Graphs

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **WN** | 0.006 (0.000) | 0.478 (0.041) | 0.418 (0.069) | 0.677 (0.007) | 0.873 (0.017) |
| **AR(1)-0.5** | 0.001 (0.000) | 0.473 (0.037) | 0.381 (0.074) | 0.623 (0.006) | 0.867 (0.019) |
| **AR(1)0.5** | 0.020 (0.001) | 0.468 (0.043) | 0.442 (0.091) | 0.761 (0.009) | 0.885 (0.021) |
| **AR(1)0.9** | 0.062 (0.002) | 0.424 (0.040) | 0.338 (0.093) | 0.786 (0.010) | 0.916 (0.026) |
| **AR(2)** | 0.076 (0.001) | 0.417 (0.036) | 0.775 (0.127) | 0.974 (0.012) | 0.863 (0.018) |
| **ARIMA** | 0.642 (0.109) | 0.216 (0.124) | 0.362 (0.217) | 0.205 (0.089) | 0.622 (0.174) |
| **ARFIMA-0.5** | 0.006 (0.000) | 0.472 (0.046) | 0.401 (0.079) | 0.645 (0.008) | 0.866 (0.022) |
| **ARFIMA0.5** | 0.050 (0.001) | 0.430 (0.050) | 0.454 (0.105) | 0.775 (0.012) | 0.867 (0.027) |
| **ARFIMA0.9** | 0.185 (0.007) | 0.330 (0.054) | 0.268 (0.102) | 0.656 (0.026) | 0.896 (0.040) |
| **ARCH** | 0.014 (0.001) | 0.363 (0.050) | 0.432 (0.094) | 0.604 (0.030) | 0.830 (0.028) |
| **GARCH1** | 0.008 (0.001) | 0.442 (0.038) | 0.458 (0.094) | 0.660 (0.009) | 0.848 (0.021) |
| **GARCH2** | 0.011 (0.001) | 0.384 (0.056) | 0.540 (0.114) | 0.605 (0.032) | 0.792 (0.034) |
| **EGARCH** | 0.010 (0.000) | 0.444 (0.043) | 0.368 (0.075) | 0.683 (0.008) | 0.883 (0.018) |
| **GJR-GARCH** | 0.081 (0.002) | 0.306 (0.037) | 0.262 (0.088) | 0.800 (0.0017) | 0.947 (0.024) |
| **SETAR** | 0.004 (0.000) | 0.452 (0.036) | 0.315 (0.060) | 0.640 (0.007) | 0.890 (0.015) |
| **INAR** | 0.017 (0.001) | 0.455 (0.110) | 0.434 (0.095) | 0.639 (0.012) | 0.893 (0.023) |
| **HMM** | 0.011 (0.001) | 0.471 (0.054) | 0.354 (0.074) | 0.676 (0.008) | 0.902 (0.019) |

Table 4.1: Table of mean values of the 100 instances of each time series model for each topological metric, resulting from NVGs. The standard deviations are presented in parentheses.

**Average Degree** The networks corresponding to the ARIMA processes are the ones that stand out the most, having the highest value (0.642). This is possibly related to the a sharp trend that these processes present. This trend implies a greater "visualization" of most of the data points to the other data points, since as the series increases and/or decreases the higher values will be able to have a line of visibility for almost all the lowest values and vice versa since the visibility is mutual. Consequently the corresponding nodes in the networks (that is, almost all nodes) will be highly connected leading to a high average degree. Similarly, the networks corresponding to the ARFIMA0.9 models stand out from the other values, although with less intensity, because these models present trends which are stochastic.

The boxplot of the values represented in figure 4.1a illustrates well that models producing smoother time series GJR−GARCH, AR(2), AR(1)0.9 and ARFIMA0.5 present higher average degree in NVG representation than models producing time series characterized by high variability.




(a) Average Degree

(b) Average Path Length

(c) Number of Communities

(d) Clustering Coefficient

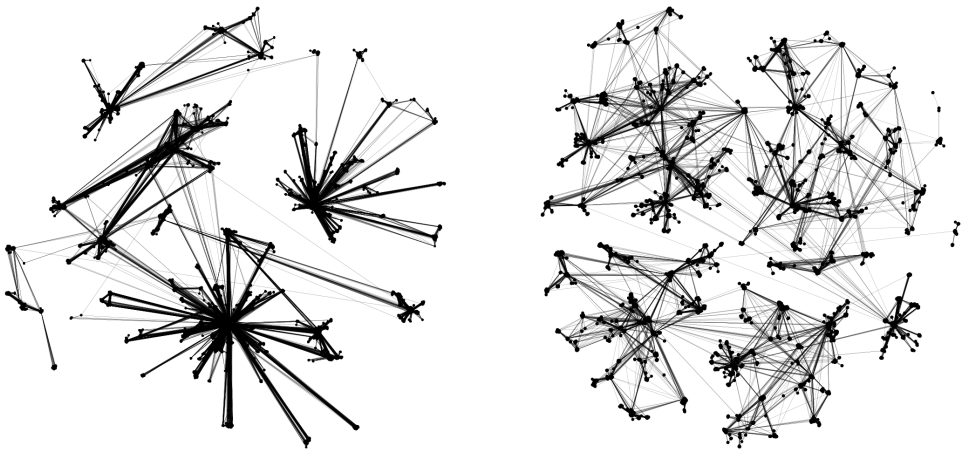
Figure 4.1: Boxplots for topological metrics of the NVGs calculated for the 100 networks of each type of model, presented through boxplots. In (a) and (d) we omitted the values for the ARIMA networks.

**Average Path Length** Also in this metric the ARIMA networks are the ones that stand out the most, with a mean value of 0.216 compared to values close to 0.4 or 0.5 for the other models. Since ARIMA has a fairly high average degree, having several nodes in the network

highly connected, we may think that this implies that the average path length is lower than those of the other types of networks. This is because there are links between almost all nodes and consequently few intermediate nodes are needed between any pairs of nodes, as compared to the other types of networks.

This metric also seems to be able to stand out in networks representing time series with some relevant peaks, namely `GJR-GARCH`, `ARFIMA0.9`, `ARCH` and `GARCH2`. The nodes of the graph corresponding to such peaks are highly connected and thus function as bridges between sets of nodes, decreasing the number of intermediate nodes required. An example of such graphs can be seen in the figure 4.2.

The boxplot of the average path length for NVG networks is represented in figure 4.1b.



(a) `ARIMA` (b) `ARFIMA0.9`

Figure 4.2: Illustration of an example of a (a) `ARIMA` type network and one of type (b) `ARFIMA0.9`, of 10000 nodes.

**Number of Communities** The number of communities stands out as being highest in the networks corresponding to the second order autoregressive models (`AR(2)`), since these processes for specific sets of parameters such as the set used in these simulations are pseudo periodic: the peak of the pseudo cycles cuts the visibility of the through. Consequently the formation of communities in the resulting networks is inevitable, since it is possible to create subsets of highly connected nodes corresponding to the data between peaks in the time series.

In contrast, the networks corresponding to the `GJR-GARCH` e `ARFIMA0.9` models stand out as networks with fewer communities. This may be related to the appearance of some (but few) sharp peaks, where in the case of `ARFIMA` these peaks correspond to changes in the trend slope direction, as well as the low variability of the observations. For the same reason these peaks limit the formation of few communities.

We might expected a small number of communities for ARIMA networks, even smaller than for ARFIMA0.9 networks, as illustrated in figure 4.2. However, the mean value in table 4.1 does not suport this presumption and the boxplot in figure 4.1c shows a very large variability in the number of communities in ARIMA networks (in agreement with the standard variation of 0.217), justifiable once again by the nonstationarity of the process.

**Clustering Coefficient** The highest value for this measure is attained by the AR(2) model, approximately 1, followed closely by GJR-GARCH, AR(1)0.9, ARFIMA0.5 and AR(1)0.5 models with values of approximately 0.8. This means that the networks with more closed triangles, which in the case of AR(2) are also related to the high number of communities explained earlier, these subgraphs being dense have several links between all the nodes in these subgraphs and, consequently, more triangles. On the other hand the ARIMA networks show the smallest value for this feature (0.205), implying the existence of few closed triangles, justified by the trending characteristics of these processes. We can interpret that it is frequent there is a link between the node $v_i$ and the node $v_j$ and a link between the node $v_j$ and the node $v_k$, that is, possible triangles, however, the existence of a link between the node $v_i$ and the node $v_k$ are less likely, since given the trend it is likely that the value $y_j$, corresponding to the node $v_j$, has a value that cuts the visibility between the values $y_i$ and $y_k$.

**Modularity** Modularity is similar for all types of networks (ranging from 0.8 to 0.9) except for those corresponding to the ARIMA model with a value of 0.622. This indicates that ARIMA networks present worse structure of internal communities when compared with the other model networks. We can conclude that this metric is not good for characterizing these different types of networks.

### 4.1.2 Horizontal Visibility Graphs

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **WN** | 0.998 (0.001) | 0.005 (0.002) | 0.223 (0.036) | 0.606 (0.004) | 0.254 (0.054) |
| **AR(1)-0.5** | 0.998 (0.001) | 0.005 (0.001) | 0.131 (0.025) | 0.559 (0.004) | 0.341 (0.052) |
| **AR(1)0.5** | 0.998 (0.001) | 0.009 (0.002) | 0.332 (0.050) | 0.699 (0.005) | 0.238 (0.058) |
| **AR(1)0.9** | 0.997 (0.001) | 0.030 (0.004) | 0.257 (0.078) | 0.791 (0.005) | 0.523 (0.070) |
| **AR(2)** | 0.997 (0.001) | 0.021 (0.002) | 0.581 (0.116) | 0.956 (0.004) | 0.127 (0.085) |
| **ARIMA** | 0.928 (0.031) | 0.539 (0.151) | 0.060 (0.026) | 0.990 (0.004) | 0.965 (0.017) |
| **ARFIMA-0.5** | 0.998 (0.001) | 0.008 (0.002) | 0.165 (0.046) | 0.584 (0.004) | 0.327 (0.070) |
| **ARFIMA0.5** | 0.996 (0.001) | 0.029 (0.004) | 0.286 (0.076) | 0.787 (0.005) | 0.429 (0.086) |
| **ARFIMA0.9** | 0.989 (0.003) | 0.103 (0.012) | 0.107 (0.045) | 0.904 (0.004) | 0.844 (0.038) |
| **ARCH** | 0.998 (0.001) | 0.009 (0.002) | 0.204 (0.041) | 0.619 (0.005) | 0.330 (0.074) |
| **GARCH1** | 0.998 (0.001) | 0.007 (0.002) | 0.214 (0.041) | 0.613 (0.004) | 0.269 (0.069) |
| **GARCH2** | 0.997 (0.001) | 0.013 (0.003) | 0.211 (0.069) | 0.618 (0.004) | 0.306 (0.097) |
| **EGARCH** | 0.998 (0.001) | 0.004 (0.001) | 0.202 (0.039) | 0.592 (0.004) | 0.248 (0.061) |
| **GJR-GARCH** | 0.997 (0.001) | 0.015 (0.002) | 0.242 (0.064) | 0.643 (0.006) | 0.504 (0.071) |
| **SETAR** | 0.998 (0.001) | 0.003 (0.001) | 0.066 (0.027) | 0.506 (0.004) | 0.573 (0.047) |
| **INAR** | 0.021 (0.008) | 0.032 (0.004) | 0.330 (0.071) | 0.034 (0.012) | 0.524 (0.054) |
| **HMM** | 0.641 (0.006) | 0.012 (0.002) | 0.230 (0.049) | 0.465 (0.006) | 0.475 (0.056) |

Table 4.2: Table of mean values of the 100 instances of each time series model for each topological metric, resulting from HVGs. The standard deviations are presented in parentheses.

**Average Degree** The mean average degree is about 1 except for INAR and HMM models with values of 0.021 and 0.641, respectively. This tells us that the INAR and HMM models have little horizontal visibility compared to the other time series models, that is, between any two points in the corresponding time series there is more frequently point(s) with higher value. The fact that INAR models have a much lower average degree may be related to the fact that these are highly correlated count time series. In these series many of the points only have visibility to their neighbors (left and right) because they have the same value. The HMM models have some similarities with the INAR models, they are quite correlated, but on the other hand they have more variability, hence the value of $\bar{k}$ of their networks is higher.

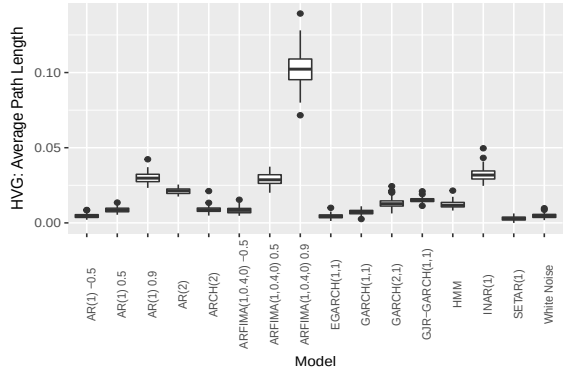
**Average Path Length** The values, varying between 0.003 and 0.032 are quite small for all networks. ARFIMA0.9 with a mean of 0.103 and specially ARIMA with a value of 0.539, stand out from the remaining. We can inevitably relate these values to the trending behavior that is mapped into a chain graph by the horizontal visibility mapping. Figure 4.3a and 4.3b illustrate 100 nodes HVG's obtained from an ARIMA and from a SETAR time series, the models that present, respectively, the highest and the lowest average path length in the set.

(a) ARIMA (b) SETAR

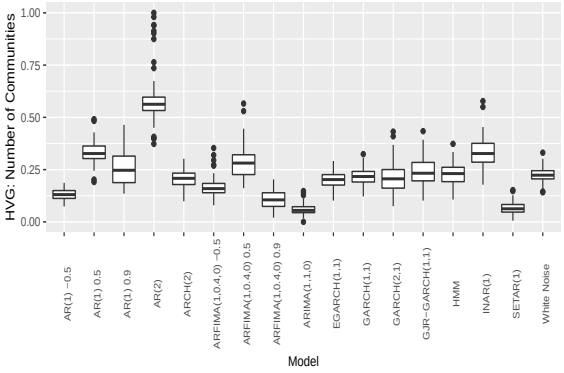
Figure 4.3: Illustration of an example of a (a) ARIMA type network and one of type (b) SETAR, of 100 nodes. The darker nodes correspond to the most connected nodes and the lightest nodes to the least connected.

The boxplots in figure 4.4a show the distribution of this measure over the 16 models (the boxplot for ARIMA was removed) indicating that the variability for ARFIMA0.9 is much larger.

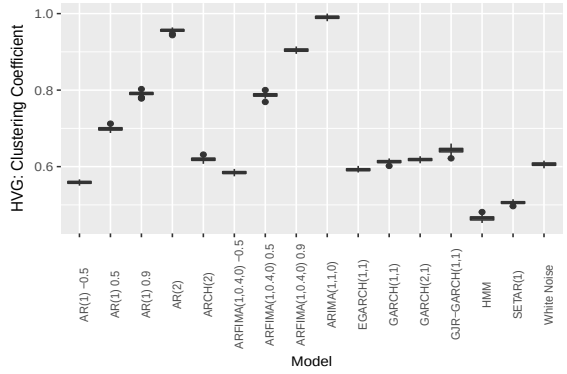
**Number of Communities** In this metric the types of networks that stand out most are AR(2), which have the highest value, and ARIMA and SETAR types, which have the lowest values. Again, AR(2) models stand out in this metric with the greatest number of communities. The SETAR models and the ARIMA models are mapped into networks with the least number of dense subgraphs, possibly due to the trend of the ARIMA models and the two regimes in the SETAR models.



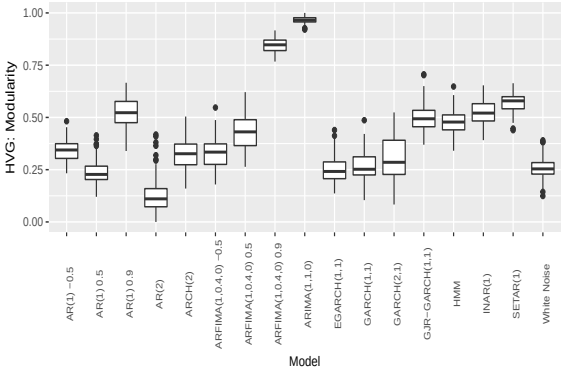

(a) Average Path Length




(b) Number of Communities




(c) Clustering Coefficient




(d) Modularity

Figure 4.4: Boxplots for the topological metrics of the HVGs calculated for the 100 networks of each type of model. In (a) we omitted the values for the ARIMA networks and in (c) the values for the INAR networks.

The boxplots in figure 4.4b indicate the presence of outliers in the values and that the networks corresponding to the ARCH, GARCH1, EGARCH and WN models show very similar distributions of the number of communities.

**Clustering Coefficient** The INAR networks have a value very close to 0, meaning that they have much less closed triangles than the other types of networks. This, as seen in the average degree metric, may be related to the fact that most points in the corresponding time series have visibility only for their two closest neighbors, this implies a large number of possible triangles, but a smaller number of closed triangles, given the lack of a line of visibility between the respective extreme neighbors. There are also similarities among models ARIMA, AR(2), ARFIMA0.9, AR(1)0.9, ARFIMA0.5 and AR(1)0.5 with values between $[0.643, 0.99]$, and the remaining networks with values between $[0.465, 0.619]$. The first group refers to the time series with low frequency and the second to the series with high frequency.

If we analyze the figure 4.4c, we see a possible relation between the values of the parameter $\phi$ in AR and ARFIMA models and this metric: the larger the value of $\phi$ the higher the number of closed triangles in the corresponding networks. We also see that the models of ARCH and GARCH family and white noise are quite similar in this metric.

**Modularity** This measure of quality of a specific division of a network is larger for `ARIMA` (0.965) and `ARFIMA0.9` type networks (0.844) and smaller for second order `AR(2)` (0.127). We can also distinguish two other groups (figure 4.4d), one with values between $[0.238, 0.341]$ and another with values between $[0.429, 0.573]$. The first group represents the networks corresponding to the `WN`, `AR(1)-0.5`, `AR(1)0.5`, `ARFIMA-0.5`, `ARCH`, `GARCH1`, `GARCH2` and `EGARCH` processes, which are time series high frequency and great variability. The second group represents the networks corresponding to the processes with different characteristics, such as lower frequency, trends, peaks and high correlation between the observations: `AR(1)0.9` , `ARFIMA0.5`, `GJR-GARCH`, `SETAR`, `INAR` and `HMM` models.

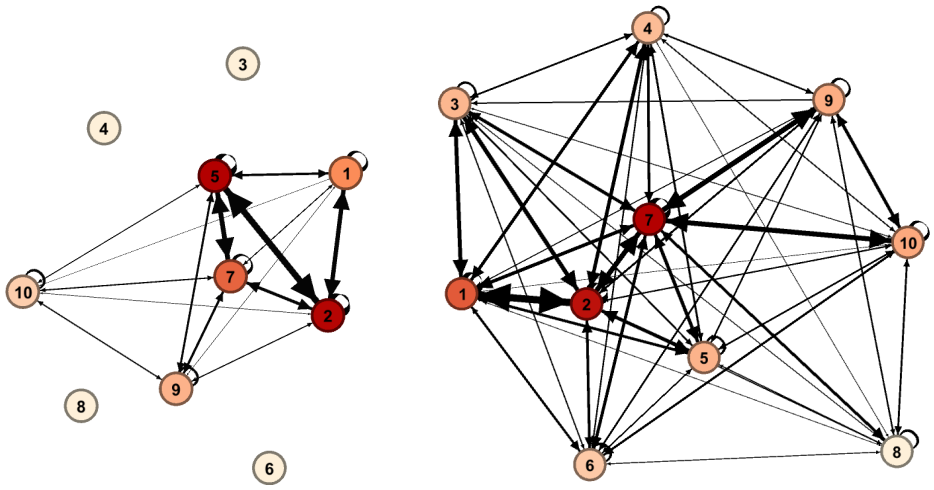
### 4.1.3 Quantile Graphs: $Q = 10$

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **WN** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.268 (0.008) |
| **AR(1)-0.5** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.169 (0.008) |
| **AR(1)0.5** | 0.000 (0.000) | 0.000 (0.000) | 0.355 (0.228) | 1.000 (0.000) | 0.476 (0.018) |
| **AR(1)0.9** | 0.000 (0.000) | 0.084 (0.010) | 0.500 (0.000) | 0.846 (0.020) | 0.920 (0.058) |
| **AR(2)** | 0.000 (0.000) | 0.052 (0.006) | 0.500 (0.000) | 0.903 (0.017) | 0.885 (0.042) |
| **ARIMA** | 0.000 (0.000) | 1.000 (0.000) | 0.005 (0.050) | 0.000 (0.000) | 0.070 (0.032) |
| **ARFIMA-0.5** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.288 (0.014) |
| **ARFIMA0.5** | 0.000 (0.000) | 0.073 (0.011) | 0.500 (0.000) | 0.864 (0.022) | 0.916 (0.055) |
| **ARFIMA0.9** | 0.000 (0.000) | 0.561 (0.076) | 0.065 (0.169) | 0.495 (0.033) | 0.668 (0.038) |
| **ARCH** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.346 (0.010) |
| **GARCH1** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.293 (0.009) |
| **GARCH2** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.318 (0.010) |
| **EGARCH** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.264 (0.008) |
| **GJR-GARCH** | 0.000 (0.000) | 0.212 (0.008) | 0.550 (0.151) | 1.000 (0.000) | 0.840 (0.090) |
| **SETAR** | 0.000 (0.000) | 0.013 (0.005) | 0.500 (0.000) | 1.000 (0.000) | 0.400 (0.012) |
| **INAR** | 0.667 (0.000) | 0.000 (0.002) | 0.000 (0.000) | 1.000 (0.000) | 0.722 (0.007) |
| **HMM** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.371 (0.009) |

Table 4.3: Table of mean values of the 100 instances of each time series model for each topological metric, resulting from 10-QGs. The standard deviations are presented in parentheses.

**Average Degree** In this type of mapping, based on a different concept, the links are weighted and consequently this metric refers to the *average weighted degree*. The values that stand out most are the values corresponding to the networks resulting from the `INAR` models, with a value of 0.667 compared to the other models that have the minimal value 0. This is expectable since the INAR model produces time series of counts, highly correlated and with little variability. In the QG, these characteristics translate into nodes with highly weighted loops links, especially between nodes that correspond to neighboring quantiles. In these graphs, some nodes are not connected to any other node, which also influences this metric.

In figure 4.5 we illustrate an example of a 10-QG for the INAR(1) model and one for the HMM model.



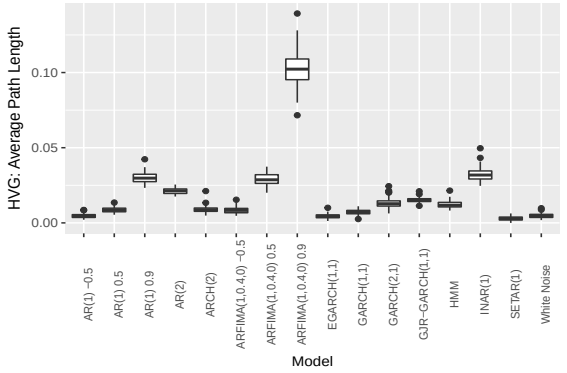

(a) `INAR` (b) `HMM`

Figure 4.5: Illustration of 10-QG networks (a) `INAR` and (b) `HMM`. The darker nodes correspond to nodes with the highest average weighted degree and the lightest nodes to the nodes with the lowest average weighted degree.

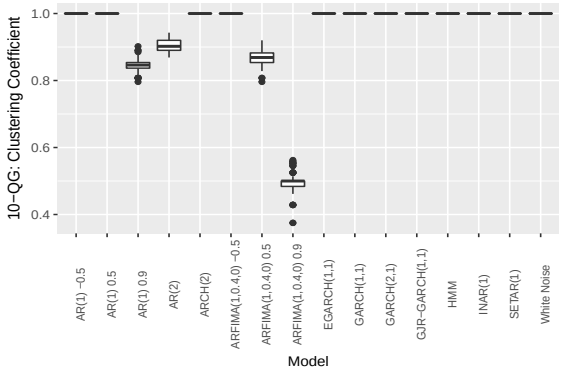
**Average Path Length** Before we perform our analysis on this metric, it should be noted that the function used to calculate the average path length of the QGs calculates the shortest paths among all the vertex pairs (in both directions for the graphics), but does not consider the weights of the links.

We can distinguish five groups of values, the highest value corresponds to the `ARIMA` networks (1), the second highest value to the `ARFIMA0.9` networks (0.561), the following to the `GJR-GARCH` networks (0.212), a set of values between $[0.013, 0.084]$ corresponds to the `SETAR`, `AR(2)`, `ARFIMA0.5` and `AR(1)0.9` networks, and the remaining networks have a minimum value of 0 (see figure 4.6a).

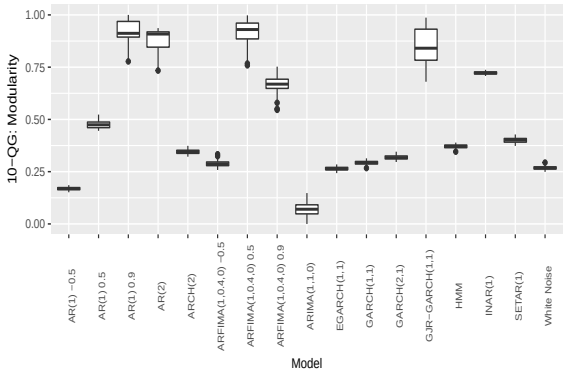
Once again, the particularity of the trend stands out (`ARIMA` and `ARFIMA0.9`), translated into a higher average path length value of the 10-QGs that will be close to chain graphs.




(a) Average Path Length




(b) Clustering Coefficient

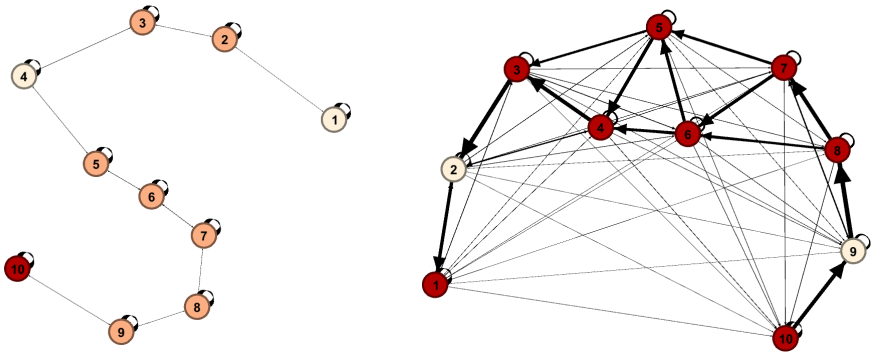



(c) Modularity

Figure 4.6: Boxplots for topological metrics of the 10-QGs calculated for the 100 networks of each model. In (a) and (b) we omitted the values for the ARIMA networks which, being too high and too low, respectively, hindered the visualization of the plots.

The GJR−GARCH models are characterized by their positive values and the existence of a few but quite sharp peaks. In this type of graphs the nodes corresponding to the intermediate quantiles will be highly connected and with loops, since GJR−GARCH series are of low variability. For this reason the average path length value is one of the highest, as it is affected by the shortest paths of the nodes corresponding to the lowest to the highest quantiles. Figure 4.7 presents an example of an ARIMA 10-QG and a GJR−GARCH 10-QG.

In the remaining two groups it is noticeable that mean average path length decreases when the smoothness of the time series decrease.




(a) ARIMA (b) GJR-GARCH

Figure 4.7: Illustration of (a) an ARIMA network and (b) a GJR-GARCH network. The darker nodes correspond to nodes with the highest average weighted degree and the lightest nodes to the nodes with the lowest average weighted degree.

**Number of Communities** The results distinguish two main groups of networks: one with more dense subgraphs and another with smaller groups. The first group refers to networks whose time series exhibit unique characteristics, namely, GJR-GARCH (positive values and the presence of sharp peak), SETAR (changes of regime and consequent presence of clusters of high or low values), ARFIMA0.5 (presence of local trends), AR(2) (periodicity), AR(1)0.9 (weak trend) and AR(1)0.5 (smooth behavior).

The second group presents the minimum value of 0, and consists essentially of the networks corresponding to the series of high variability and discrete valued. We also observed a value of 0.005 for the ARIMA type and 0.065 for the ARFIMA0.9 type, which correspond to the series with the highest trending behavior.

**Clustering Coefficient** The smaller values for the clustering coefficient are attained by ARIMA and ARFIMA0.9, which again has to do with their chain structure of the networks, leading to existence of many triangles but few closed triangles. From the table and figure 4.6b (the boxplot for ARIMA has been removed) it is clear that the the stronger the trending behavior the smaller the clustering coefficient.

**Modularity** As expected, the quality of a specific division of ARIMA networks is very low given its structure in chain. We can also distinguish three groups (figure 4.6c), one with a very high quality, namely AR(1)0.9, ARFIMA0.5, AR(2), GJR-GARCH, INAR and ARFIMA0.9 networks; followed by a group of relative quality, the networks AR(1)0.5, SETAR and HMM; and finally a group of lesser quality division of networks in communities, ARCH, GARCH2, GARCH1, ARFIMA-0.5, WN, EGARCH e AR(1)-0.5. The groups aggregate series with decreasing smoothness.

### 4.1.4 Quantile Graphs: $Q = 50$

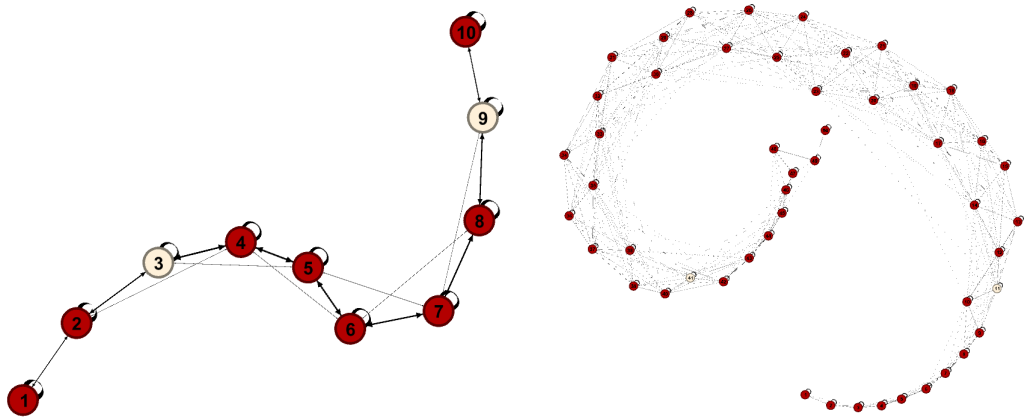
| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **WN** | 0.000 (0.000) | 0.001 (0.000) | 0.193 (0.077) | 1.000 (0.000) | 0.036 (0.008) |
| **AR(1)-0.5** | 0.000 (0.000) | 0.005 (0.000) | 0.000 (0.000) | 0.972 (0.004) | 0.005 (0.003) |
| **AR(1)0.5** | 0.000 (0.000) | 0.005 (0.000) | 0.160 (0.047) | 0.971 (0.003) | 0.295 (0.031) |
| **AR(1)0.9** | 0.000 (0.000) | 0.031 (0.001) | 0.313 (0.099) | 0.799 (0.005) | 0.706 (0.026) |
| **AR(2)** | 0.000 (0.000) | 0.024 (0.001) | 0.267 (0.099) | 0.829 (0.004) | 0.634 (0.022) |
| **ARIMA** | 0.000 (0.000) | 0.945 (0.059) | 0.411 (0.150) | 0.151 (0.132) | 0.317 (0.073) |
| **ARFIMA-0.5** | 0.000 (0.000) | 0.001 (0.000) | 0.156 (0.071) | 1.000 (0.001) | 0.050 (0.019) |
| **ARFIMA0.5** | 0.000 (0.000) | 0.029 (0.002) | 0.307 (0.104) | 0.808 (0.008) | 0.685 (0.032) |
| **ARFIMA0.9** | 0.000 (0.000) | 0.160 (0.017) | 0.651 (0.155) | 0.692 (0.008) | 0.957 (0.016) |
| **ARCH** | 0.000 (0.000) | 0.002 (0.000) | 0.257 (0.086) | 0.996 (0.002) | 0.125 (0.016) |
| **GARCH1** | 0.000 (0.000) | 0.001 (0.000) | 0.173 (0.062) | 1.000 (0.001) | 0.043 (0.012) |
| **GARCH2** | 0.000 (0.000) | 0.001 (0.000) | 0.233 (0.075) | 0.999 (0.001) | 0.078 (0.014) |
| **EGARCH** | 0.000 (0.000) | 0.002 (0.000) | 0.126 (0.074) | 0.999 (0.001) | 0.032 (0.012) |
| **GJR-GARCH** | 0.000 (0.000) | 0.059 (0.001) | 0.327 (0.082) | 0.839 (0.008) | 0.653 (0.023) |
| **SETAR** | 0.000 (0.000) | 0.016 (0.000) | 0.181 (0.067) | 0.946 (0.004) | 0.194 (0.026) |
| **INAR** | 0.999 (0.015) | 0.002 (0.001) | 0.000 (0.000) | 0.986 (0.024) | 0.418 (0.004) |
| **HMM** | 0.270 (0.009) | 0.001 (0.000) | 0.143 (0.000) | 0.999 (0.003) | 0.182 (0.019) |

Table 4.4: Table of mean values for each topological metric computed from 50-QGs. The corresponding standard deviations are presented in parentheses.

**Average Degree** Similarly to 10-QGs, the networks with the highest average degree are the networks of type INAR with a value of 0.999.

We note that with the increase in the number of quantiles (and consequently, nodes) the value for the HMM type networks, 0.270, also stands out in contrast to the other networks (with value 0). From this we can conclude that the increase in the number of quantiles can capture similarities in the time series corresponding to the networks of the type HMM has with those of the type INAR that were not captured with the 10 nodes.

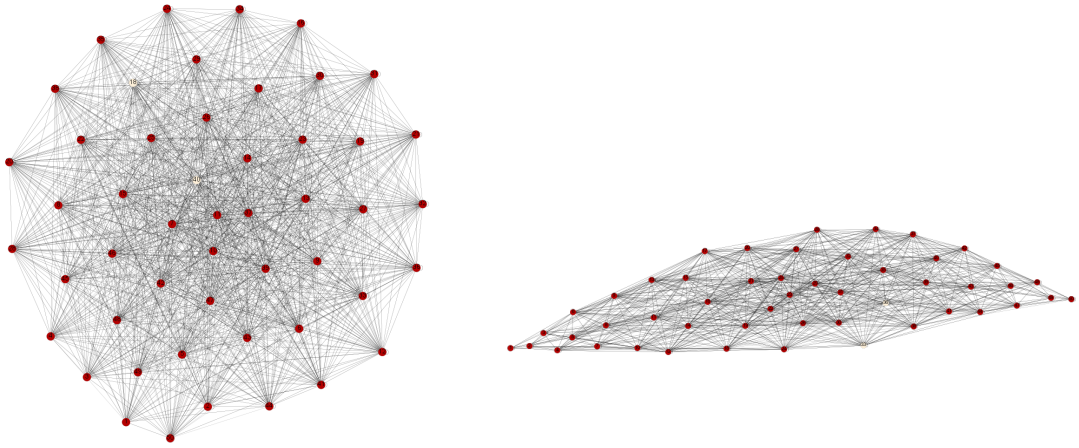
**Average Path Length** The average path length results are similar to those obtained for QGs of 10 nodes. However, the value for ARFIMA0.9 is much smaller than for 10-QG, meaning that there is a smaller number of intermediate nodes in a path between any two nodes in the network. This is due to the fact that the time series ARFIMA0.9 has local trends, different from the overall trend of the ARIMA series. Therefore, if we increase the quantity of quantiles that divides the support of the series, the variations among different quantiles will be captured by the resulting networks and, consequently, there will be nodes with connections to more nodes. This difference can be seen in the example shown in figure 4.8 which presents a 10-QG and a 50-QG for an ARFIMA0.9 series.




(a) ARFIMA0.9 (10-QG) (b) ARFIMA0.9 (50-QG)

Figure 4.8: Illustration of an example of a (a) ARFIMA0.9 type 10-QG and one of type 50-QG of (b) GJR-GARCH. The darker nodes correspond to nodes with the highest average weighted degree and the lightest nodes to the nodes with the lowest average weighted degree.

Once again, networks representing smoother series present higher average path length. An illustration is provide in figure 4.9 which represents a network from an AR(1)-0.5 characterized by high frequency variability and a network from ARFIMA0.9, characterized by local trends.



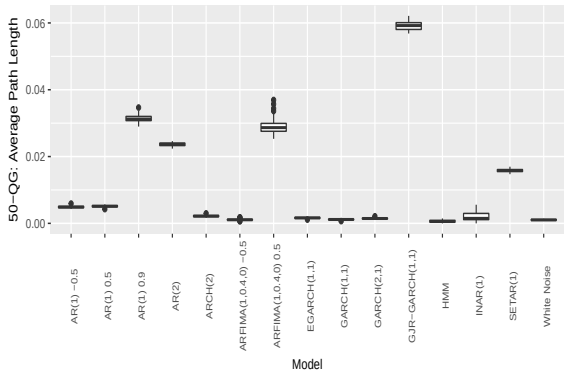
(a) AR(1)-0.5 (50-QG) (b) AR(1)0.9 (50-QG)

Figure 4.9: Illustration of an example of a (a) AR(1)-0.5 type network and one of type (b) AR(1)0.9. The darker nodes correspond to nodes with the highest average weighted degree and the lightest nodes to the nodes with the lowest average weighted degree.

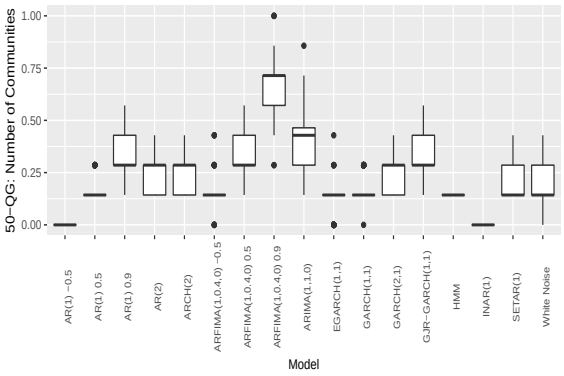
**Number of Communities** The distribution of this metric is highly asymmetric for some of the models, presenting also a larger variability. The largest mean number of communities occurs for ARFIMA0.9 networks while AR(1)-0.5 and INAR networks present the minimum possible value (see figure 4.10). We also noticed that the value of the ARIMA networks increased when compared to the previous mapping. An explanation is that the trend implies the formation of QGs of 50 nodes with the chain structure and consequently this structure will be divided in some communities as the corresponding time series increases (or decreases) will have larger connections between the higher (or lower) quantiles (corresponding nodes) and smaller connections between more distant quantiles. The same comment applies to ARFIMA0.9.

**Clustering Coefficient** The mean values have decreased in comparison to 10-QGs but the overall conclusions are the same as indicated by the plots in figure 4.10c.

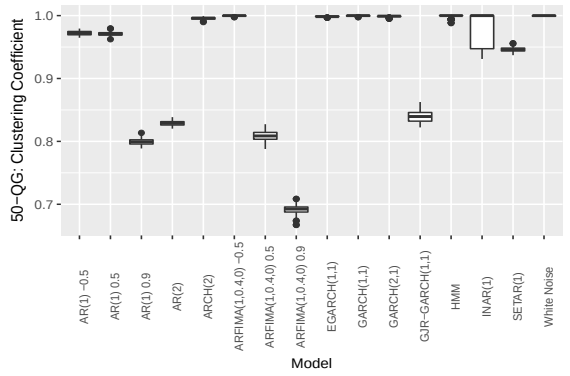
**Modularity** The difference between the modularity obtained from the 10-QGs and the 50-QGs resides essentially in a decrease in the quality of a specific division of the network, since the number of nodes increases significantly. The ARIMA type networks are the exception with an increases from 0.070 to 0.317. However this decrease is not in the same proportion for all types of networks, which leads us to distinguish different groups of values (see figure 4.10d) compared to previous ones. In particular, the highest values are obtained by the ARFIMA0.9, AR(1)0.9, ARFIMA0.5, GJR-GARCH and AR(2) networks, followed by the values obtained by the INAR, ARIMA and AR(1)0.5 networks, after the values of SETAR and HMM, and finally the best values obtained by ARCH, GARCH2, ARFIMA-0.5, GARCH1, WN, EGARCH and AR(1)-0.5 networks. The last group can be explained by the fact that the networks represent the processes have the highest frequency of oscillations, especially the AR(1)-0.5 model given its negative regression



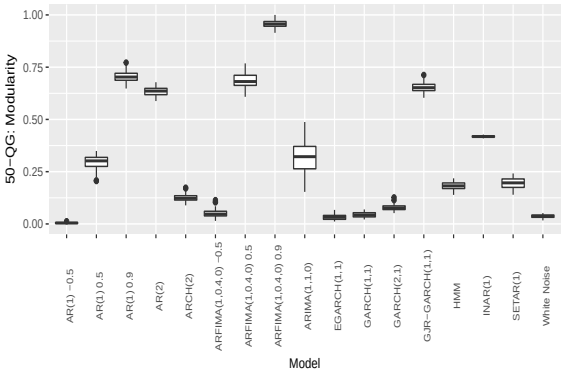

(a) Average Path Length




(b) Number of Communities




(c) Clustering Coefficient




(d) Modularity

Figure 4.10: Boxplots for topological metrics of the 50-QGs calculated for the 100 networks for each model. We omitted the plots for `ARIMA` and `ARFIMA0.9` networks in (a) and for the `ARIMA` networks in (c).

coefficient and hence the type of network with lower modularity value. This high frequency translates into an oscillation between the different quantiles more or less similar which makes the division in specific communities is relatively bad.

### 4.1.5 Quantile Graphs: $Q = 100$

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **WN** | 0.000 (0.000) | 0.011 (0.000) | 0.176 (0.075) | 0.858 (0.005) | 0.061 (0.011) |
| **AR(1)-0.5** | 0.000 (0.000) | 0.012 (0.000) | 0.000 (0.000) | 0.789 (0.005) | 0.004 (0.002) |
| **AR(1)0.5** | 0.000 (0.000) | 0.012 (0.000) | 0.126 (0.048) | 0.801 (0.004) | 0.261 (0.025) |
| **AR(1)0.9** | 0.000 (0.000) | 0.022 (0.000) | 0.252 (0.081) | 0.721 (0.004) | 0.641 (0.021) |
| **AR(2)** | 0.000 (0.000) | 0.018 (0.000) | 0.196 (0.048) | 0.731 (0.003) | 0.566 (0.017) |
| **ARIMA** | 0.000 (0.000) | 0.735 (0.124) | 0.561 (0.120) | 0.402 (0.119) | 0.523 (0.096) |
| **ARFIMA-0.5** | 0.000 (0.000) | 0.011 (0.000) | 0.185 (0.101) | 0.855 (0.005) | 0.070 (0.014) |
| **ARFIMA0.5** | 0.000 (0.000) | 0.021 (0.001) | 0.222 (0.068) | 0.724 (0.004) | 0.619 (0.025) |
| **ARFIMA0.9** | 0.000 (0.000) | 0.089 (0.009) | 0.557 (0.115) | 0.686 (0.004) | 0.956 (0.015) |
| **ARCH** | 0.000 (0.000) | 0.011 (0.000) | 0.205 (0.068) | 0.837 (0.005) | 0.111 (0.015) |
| **GARCH1** | 0.000 (0.000) | 0.011 (0.000) | 0.192 (0.118) | 0.855 (0.005) | 0.058 (0.011) |
| **GARCH2** | 0.000 (0.000) | 0.011 (0.000) | 0.182 (0.067) | 0.848 (0.004) | 0.075 (0.013) |
| **EGARCH** | 0.000 (0.000) | 0.011 (0.000) | 0.180 (0.174) | 0.852 (0.005) | 0.046 (0.018) |
| **GJR-GARCH** | 0.000 (0.000) | 0.037 (0.001) | 0.233 (0.052) | 0.550 (0.006) | 0.583 (0.017) |
| **SETAR** | 0.000 (0.000) | 0.016 (0.000) | 0.117 (0.047) | 0.785 (0.005) | 0.166 (0.023) |
| **INAR** | 1.000 (0.000) | 0.002 (0.001) | 0.000 (0.000) | 0.954 (0.025) | 0.363 (0.003) |
| **HMM** | 0.327 (0.002) | 0.001 (0.000) | 0.091 (0.000) | 0.981 (0.007) | 0.163 (0.017) |

Table 4.5: Table of mean values of the 100 instances of each time series model for each topological metric, resulting from 100-QGs. The corresponding standard deviations are presented in parentheses.

**Average Degree** As in the QGs with 50 nodes, the networks with the highest average degree are the networks of type INAR with a value of 0.967 and HMM with a value of 0.335, the possible reasons for this remain the same.

**Average Path Length** The average path length results are also similar to those obtained for 50-QGs. However, the higher values decrease a bit due to the increase in the quantity of quantiles that consequently leads to a greater transition between different quantiles which translates into the network as a greater connection between different nodes.

**Number of Communities** The number of communities results are also similar to those obtained for QGs of 50 nodes and the possible reasons for this as well.

**Clustering Coefficient** In this type of graph the values vary in the range of $[0.402, 0.981]$, the lowest values refer to ARIMA and GJR−GARCH networks, and the highest values to the HMM and INAR networks. As such, this metric seems to highlight the most evident features of the series that these networks represent, namely, correlations between the discrete values of the HMM and INAR models, the trend and low variability of the ARIMA models and the sharp peaks and low variability of the GJR−GARCH. We can also distinguish a group of values between $[0.683, 0.801]$ and another between $[0.837, 0.858]$.

**Modularity** The modularity results are similar to those obtained for QGs of 50 nodes and the possible reasons for this as well.

### 4.1.6 Summary

The main conclusions that we can draw are as follows.

NVGs appear to be highlighting trending behavior of the series, such as that found in ARIMA, ARFIMA0.9, AR(1)0.9 and ARFIMA0.5. This conclusion is further supported by the results presented in the table C.1 in the appendix C which contains the values for the metrics calculated for time series with different types of trend, including deterministic trends (models denoted by D_Trend). NVGs also highlight series with peaks or special variability, such as GJR−GARCH and AR(2) models. The average degree and clustering coefficient of the NVGs tend to increase with increasing values of $\phi_1$ in AR and ARFIMA models, behavior which may be related to increased autocorrelation. On the other hand, the average path length tends to decrease with higher autocorrelation. It is worth noting that EGARCH, GARCH1 and GARCH2 have rather different number of communities, the first has the lowest number of dense subgraphs and the last the largest.

The HVGs, however, seem to be able to distinguish discrete valued time series, INAR and HMM, using the average degree. The SETAR model also stands out with number of communities similar to the ARIMA models. Some metrics also seem to distinguish networks corresponding to series with larger variability of networks corresponding to series of smaller variability. Clustering coefficients seem to increase with increasing value of $\phi_1$ in AR and ARFIMA series, and are similar for the ARCH and GARCH family and white noise models. Thus we can conclude that the clustering coefficient in HVG's is related to the autocorrelation.

Regarding 10-QGs networks, the results indicate that number of communities characterize well GJR-GARCH models, models leading to cycles and smooth time series, AR(1)0.5, AR(1)0.9, AR(2), ARFIMA0.5 and ARFIMA0.9. The clustering coefficient can capture the different trends, ARIMA, ARFIMA0.9 and AR(1)0.9. While modularity seems to distinguish trend series, low frequency series and high frequency series.

It is interesting to note that the number of communities of 50-QGs highlights AR(1)-0.5 which had not been distinguished by the other measures. The 100-QGs capture substantially the same characteristics as the 50-QGs.

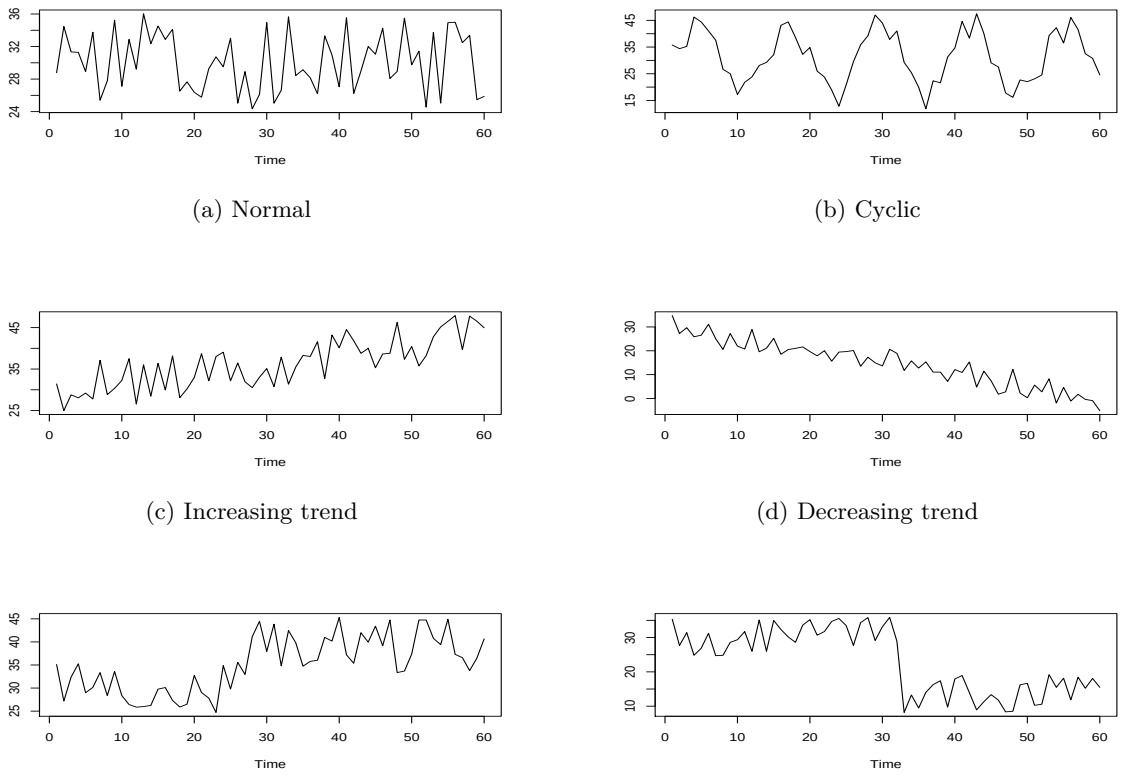
We can also draw some conclusions about the differences of the values related to the metrics analyzed when applied to networks corresponding to time series of different lengths. The results of this small study are presented in appendix A and appendix B. In the former, the results from mapping time series of length $T = 1000$ are presented in tables organized similarly to tables 4.1 to 4.5. In the latter, a graphical display allows to compare the mean measure for three values of $T = \{100, 1000, 10000\}$. Overall the topological measures provide the same information for the three sample sizes with the following exceptions. In NVG, modularity presents different information for $T = 100$ and AR(1)0.9, ARFIMA0.9, GJR-GARCH; $T = 1000$ and ARFIMA0.9. In HVG, the number of communities presents different information for $T = 100$ and ARIMA; $T = 1000$ and GARCH2. Modularity in $T = 100$ and AR(1)-0.5, AR(2), GARCH1, INAR, HMM; in $T = 1000$ and INAR, HMM. In QGs there are notorious differences among the three sample sizes for 50-QGs and 100-QGs, indicating that the number of quantiles must be set depending on the sample size as well as on the range of the support of the data.

## 4.2 *Synthetic Control* Time Series

In this section we repeat the above analysis for a public set of time series, designated by *synthetic control* data. The dataset is available for download at http://archive.ics.uci.edu/ml/machine-learning-databases/synthetic_control-mld/. This dataset contains 600 examples of control charts, where each set of 100 charts corresponds to a different type of pattern, and the length ($T$) of each time series is 60. Control chart patterns are time series that show the level of a machine parameter plotted against time. They are artificially generated by the following six equations, each representing a different type of pattern [4].

- **Normal pattern:** $y_t = m + rs$, where $m = 30$, $s = 2$ and $r$ is a random number between $[-3, 3]$,
- **Cyclic pattern:** $y_t = m + rs + a\sin(2\pi(\frac{t}{T}))$, where $a, T \in [10, 15]$, $a$ is the amplitude of cyclical variations, and $T$ is the period of one cycle,
- **Increasing trend:** $y_t = m + rs + gt$, where $g \in [0.2, 0.5]$ is the gradient of a increasing trend pattern (or a decreasing trend pattern),
- **Decreasing trend:** $y_t = m + rs - gt$,
- **Upward shift:** $y_t = m + rs + kx$, where $x \in [7.5, 20]$ is the magnitude of the shift and $k$ indicates the shift position (if $k = 0$ the shift is before time $t$ and if $k = 1$ the shift is after this time),
- **Downward shift:** $y_t = m + rs - kx$.

In the figure 4.11 we can see an example of each of these six types of patterns.




(a) Normal (b) Cyclic

(c) Increasing trend (d) Decreasing trend

(e) Upward shift (f) Downward shift

Figure 4.11: Examples of *synthetic control* time series, namely, a normal pattern, a cyclic pattern, an increasing trend, a decreasing trend, an upward shift, and a downward shift.

The distinction between different types of patterns is essential, since, except for normal patterns, they show that the process being monitored is not working correctly and requires adjustments depending on the type of pattern. This leads us to the usual task of classification or characterization of time series, which is not always an easy task.

We analyse this set of time series using the same procedure as before. The QGs networks are constructed with 3, 6 and with 10 nodes due to the length of the time series. The results are summarized in tables 4.6 to 4.10.

### 4.2.1 Natural Visibility Graphs

| Classes | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **Normal** | 0.280 (0.205) | 0.321 (0.129) | 0.441 (0.145) | 0.333 (0.199) | 0.585 (0.138) |
| **Cyclic** | 0.271 (0.209) | 0.327 (0.137) | 0.439 (0.163) | 0.339 (0.206) | 0.599 (0.165) |
| **Increasing trend** | 0.268 (0.193) | 0.358 (0.191) | 0.449 (0.160) | 0.351 (0.203) | 0.607 (0.158) |
| **Decreasing trend** | 0.269 (0.204) | 0.340 (0.168) | 0.437 (0.153) | 0.333 (0.199) | 0.603 (0.167) |
| **Upward shift** | 0.266 (0.210) | 0.349 (0.161) | 0.441 (0.153) | 0.342 (0.223) | 0.605 (0.186) |
| **Downward shift** | 0.254 (0.189) | 0.334 (0.155) | 0.466 (0.149) | 0.316 (0.212) | 0.597 (0.164) |

Table 4.6: Mean values of the topological metrics of the 100 NVGs corresponding to the 100 instances of each pattern of *synthetic control* time series. The corresponding standard deviations are presented in parentheses.

The values obtained for each one of the topological metrics extracted from the NVGs are quite similar among the different time series patterns. If we rely only on this type of mapping and in this particular dataset we could conclude that NVGs are not useful for capturing the different components of the time series. However, we can not fail to take into account the fact that all the series have a common component as well as the small sample size $T = 60$ which may be considered too small for the NVGs, as the figure in appendix B.1 suggests. This also makes sense because the NVGs are, as analyzed in the literature, graphs of "global" visibility appropriate to capture peaks that do not exist in these time series. These results are also in accordance with the results in the literature which state that , as mentioned in section 3.1, NVG is not capable of distinguishing nonstationary time series with increasing trend from series with decreasing trend.

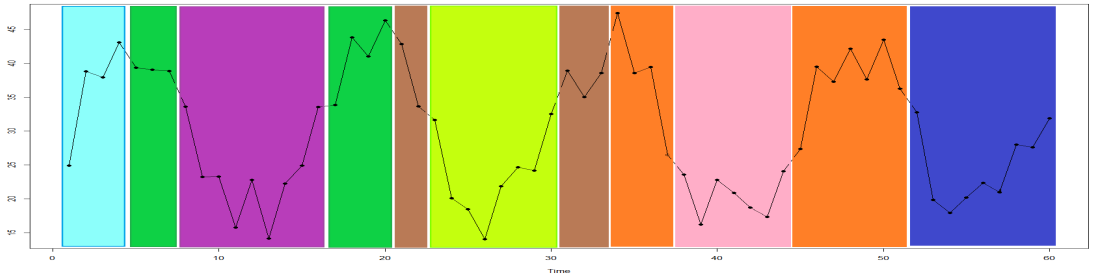
### 4.2.2 Horizontal Visibility Graphs

| Classes | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **Normal** | 0.818 (0.115) | 0.137 (0.074) | 0.396 (0.158) | 0.295 (0.113) | 0.354 (0.149) |
| **Cyclic** | 0.756 (0.112) | 0.350 (0.079) | 0.459 (0.151) | 0.718 (0.125) | 0.541 (0.108) |
| **Increasing trend** | 0.432 (0.133) | 0.521 (0.131) | 0.368 (0.120) | 0.566 (0.123) | 0.730 (0.116) |
| **Decreasing trend** | 0.421 (0.154) | 0.541 (0.150) | 0.378 (0.132) | 0.596 (0.112) | 0.737 (0.123) |
| **Upward shift** | 0.692 (0.132) | 0.299 (0.129) | 0.380 (0.159) | 0.409 (0.122) | 0.494 (0.142) |
| **Downward shift** | 0.722 (0.122) | 0.291 (0.113) | 0.358 (0.129) | 0.407 (0.111) | 0.499 (0.137) |

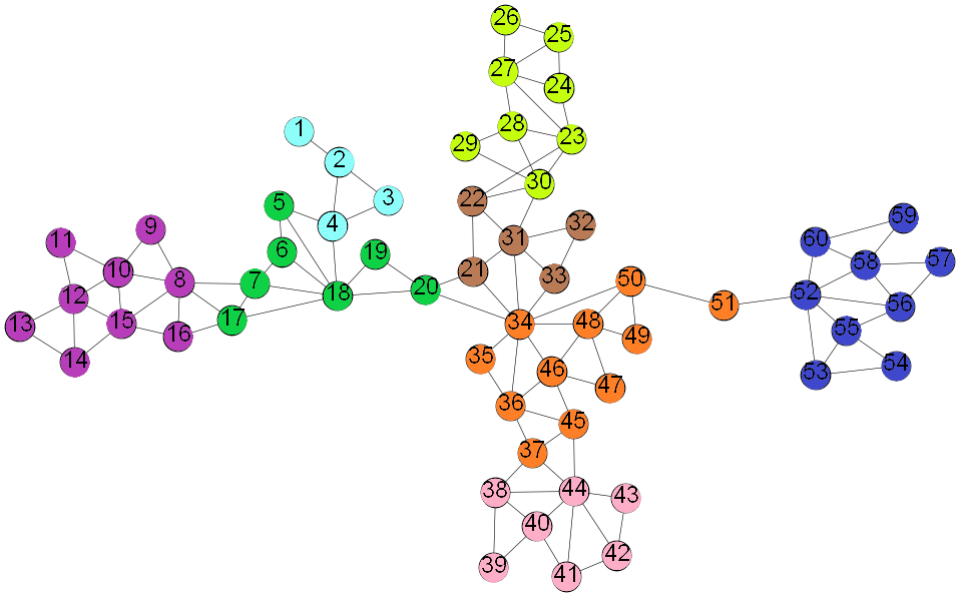
Table 4.7: Mean values of the topological metrics of the 100 HVGs corresponding to the 100 instances of each pattern of *synthetic control* time series. The respective standard deviation are presented in parentheses.

Unlike the results obtained for the NVGs, the topological metrics of the HVGs are relevant for the distinction of some of the patterns under analysis. We can verify that the trend pattern (increasing and decreasing) is manifested in the lower average degree of the resulting graphs, and consequently a higher value of average path length, since the networks are less connected and therefore the length of the path is more prone to needing more intermediate nodes than the other networks whose connections are more intense. Also the modularity value stands out for this standard.

Another important value is the number of communities in networks that represent the cyclic time series. This result is expectable because the HVG are "local" visibility graphs and are therefore more sensitive to the oscillations of time series than the NVGs (the "global" version). The figure 4.12 represents one of the cyclic time series and the corresponding HVG. We can observe four cycles in the time series and eight highly connected communities in the HVG: four corresponding to the four peaks and four corresponding to the four troughs. We also notice evidence of a pattern corresponding to these communities throughout the cycles of the time series, that is, the nodes corresponding to the closest neighbors on the right side of the peak of one cycle are densely connected to the nodes corresponding to closest neighbors on the left side of the peak of the following cycle, thus forming a community. The throughs of the cycles are densely connected forming another community. This result is in line with the statement that HVG are subgraph of NVG that can capture small particularities that are masked by the NVG. This is a good clue that we intend to explore in future work.




(a) Time serie of cyclic pattern



(b) HVG

Figure 4.12: Illustration of one of the time series of cyclic pattern and the respective HVG. The different colors represent (b) sets of dense subgraphs (communities) in the HVG, and (a) the corresponding sets of point in the time series.

The clustering coefficient also seems to be particular for the cyclic patterns, since given the cyclic variations it is probable that there are large amounts of closed triangles, the opposite happens in the normal patterns. It can still make a small distinction from the other patterns.

The modularity values clearly distinguish the trend patterns of the other patterns.

### 4.2.3 Quantile Graphs: $Q = 3$

| Classes | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **Normal** | 0.17 (0.378) | 0.000 (0.000) | 1 (0.000) | 1.00 (0.000) | 0.674 (0.206) |
| **Cyclic** | 0.05 (0.219) | 0.745 (0.314) | 1 (0.000) | 0.44 (0.499) | 0.832 (0.118) |
| **Increasing trend** | 0.16 (0.368) | 0.545 (0.439) | 1 (0.000) | 0.57 (0.498) | 0.806 (0.205) |
| **Decreasing trend** | 0.14 (0.349) | 0.640 (0.403) | 1 (0.000) | 0.50 (0.503) | 0.772 (0.231) |
| **Upward shift** | 0.26 (0.441) | 0.628 (0.325) | 1 (0.000) | 0.60 (0.492) | 0.916 (0.107) |
| **Downward shift** | 0.16 (0.368) | 0.653 (0.312) | 1 (0.000) | 0.59 (0.494) | 0.929 (0.090) |

Table 4.8: Mean values of the topological metrics of the 100 QGs of 3 quantiles corresponding to the 100 instances of each pattern of *synthetic control* time series. The respective standard deviation are presented in parentheses.

Although these graphs have very few nodes, 3, they seem to capture some pertinent information. The average degree seems to distinguish cyclical patterns (as the smallest 0.05) and upward shift patterns (as the largest value 0.26), the remaining values range from 0.14 to 0.17. This may be an important case because we have not yet been able to distinguish upward shift from downward shift, however this difference is not very pronounced (0.10). Within about the same range of difference, we could distinguish increasing trend (0.545) from downward trend (0.64) using average path length. However, these differences are too small for us to consider a good result.

The average path length of normal pattern is 0.0, a value much lower than that for the other patterns. On the other hand, the clustering coefficient for the normal pattern attains the highest value possible, 1. This means that the 3-QGs corresponding to the normal patterns are a 3-clique, that is, a graph of 3 nodes with links between all of them.

The number of communities is 1 for all types of networks, since they have only 3 nodes. And the networks formed by time series with shift have the highest values of modularity and the normal type networks have the lowest values.

### 4.2.4 Quantile Graphs: $Q = 6$

| Classes | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **Normal** | 0.94 (0.239) | 0.122 (0.053) | 0.10 (0.302) | 0.970 (0.062) | 0.260 (0.083) |
| **Cyclic** | 0.71 (0.456) | 0.403 (0.085) | 0.87 (0.338) | 0.532 (0.113) | 0.573 (0.054) |
| **Increasing trend** | 0.81 (0.394) | 0.411 (0.172) | 0.73 (0.446) | 0.539 (0.180) | 0.567 (0.114) |
| **Decreasing trend** | 0.86 (0.349) | 0.417 (0.155) | 0.80 (0.402) | 0.512 (0.161) | 0.574 (0.089) |
| **Upward shift** | 0.67 (0.473) | 0.442 (0.117) | 0.98 (0.141) | 0.507 (0.152) | 0.698 (0.142) |
| **Downward shift** | 0.66 (0.476) | 0.439 (0.107) | 0.96 (0.197) | 0.496 (0.138) | 0.719 (0.142) |

Table 4.9: Mean values of the topological metrics of the 100 QGs of 6 quantiles corresponding to the 100 instances of each pattern of *synthetic control* time series. The corresponding standard deviations are presented in parentheses.

The average degree values range from 0.66 to 0.94, where the networks corresponding to shift series have the lowest average degree (0.66 and 0.67), the trend series have more connections between the nodes (0.81 and 0.86), the networks corresponding to the normal series are even more nodes connected (0.94) and those corresponding to the cyclical series show average value of 0.71. The average degree with a higher number of nodes seems to distinguish almost all types of networks, although the difference values are not very high.

The next three metrics ($\bar{d}$, $S$ and $C$) only differentiate networks corresponding normal patterns. Modularity presents higher values for the networks corresponding to the shift patterns, reflecting the quality of the division of the networks into two subgraphs: one corresponding to the quantiles before the shift and another to the quantiles after the shift. In particular the networks corresponding to normal patterns have a very low value, since as the value of $\bar{k}$ indicates, they are quite connected.

### 4.2.5 Quantile Graphs: $Q = 10$

| Classes | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **Normal** | 0.04 (0.197) | 0.070 (0.038) | 0.347 (0.183) | 0.586 (0.123) | 0.261 (0.104) |
| **Cyclic** | 0.00 (0.000) | 0.339 (0.089) | 0.343 (0.057) | 0.404 (0.129) | 0.630 (0.051) |
| **Increasing trend** | 0.00 (0.000) | 0.370 (0.188) | 0.410 (0.170) | 0.392 (0.134) | 0.611 (0.148) |
| **Decreasing trend** | 0.00 (0.000) | 0.402 (0.182) | 0.430 (0.159) | 0.411 (0.139) | 0.647 (0.118) |
| **Upward shift** | 0.00 (0.000) | 0.457 (0.161) | 0.343 (0.074) | 0.572 (0.183) | 0.772 (0.151) |
| **Downward shift** | 0.02 (0.141) | 0.442 (0.167) | 0.340 (0.067) | 0.576 (0.147) | 0.771 (0.148) |

Table 4.10: Mean values of the topological metrics of the 100 QGs of 10 quantiles corresponding to the 100 instances of each pattern of *synthetic control* time series. The corresponding standard deviation are presented in parentheses.

In 10-QGs the average degree values are approximately 0 for all, which means that most of your nodes are weakly connected. Normal patterns are distinguishable using average path length and modularity, both with the lowest mean values of the set. Also some distinction can be observed between the shift patterns and the trend and cyclical patterns.

The number of communities is higher for the trend patterns. This is explained by the quantity of quantiles (10) and the existence of a trend (increasing or decreasing): the connections between the nodes will be in abundance for the loops but scarce or inexistent between nodes that represent non adjacent quantiles. Thus this connection structure will translate into small dense subgraphs, which will be more abundant than in the other networks that are more internally connected.

Finally, clustering coefficient leads to two groups: one containing the normal and shift patterns and the other the trend and cycle patterns. The normal and shift paterns are really similar if we exclude the shift but the shift is not picked by this metric. The trend and cycle patterns also have a similar value since the nodes corresponding to adjacent quantiles are grouped together.

### 4.2.6 Summary

From the analyzed dataset we can conclude that the NVGs are not good at capturing the special properties of small time series, namely the presence of cycles, trend and the existence of shifts in a series. However, this result can not be considered conclusive because we are dealing with a very particular dataset of time series of quite small length, and as analyzed before, B.1, the metrics are affected by small $T$.

On the other hand, the results obtained for the HVGs have shown to be very promising. We are able to verify differences in the average degree ($\bar{k}$) and modularity ($Q$) metrics for trend time series, but not between the different slopes of this trend. This result was already expected. We have also been able to speculate a possible relation between the community number ($S$) of the HVGs corresponding to the cyclic patterns and the cycles that the time series present.

We could also conclude that QGs are also quite promising for pattern distinctions, though again they are not perfect. The average degree with a higher number of nodes (6) seems to distinguish almost all types of networks, although the differences are not very high. The best quantity of quantiles is a very difficult question to answer, because it depends of the size of the time series and interval values of their observations.

# Chapter 5

# Clustering using Topological Features

We will now turn to the experimental analysis of our work. Our goal is to show that each proposed mapping method has a meaningful contribution to the time series clustering.

The purpose of clustering analysis is to discover the natural groupings of a set of patterns, points, or objects. It consists of the empirical formation of groups of objects, denominated *clusters*, with high intra-cluster similarity and low inter-cluster similarity. That is, given a representation of $n$ objects, the goal is to find $k$ clusters based on a measure of similarity, so that the similarities between objects in the same cluster are high, whereas the similarities between objects in different clusters are low [31].

When clustering time series, we can distinguish between two main categories, the one that performs clustering on a set of time series with the purpose of grouping them in different clusters, and the one that performs clustering on "windows" of a single time series with the objective of finding similarities and differences between different windows of time. Here we focus on the first category.

One of the major problems in time series clustering analysis is the choice of a relevant metric to perform grouping. According to the literature, nonparametric approaches that look only at a measure of similarity, e.g., Euclidean distance, autocorrelation, cepstrum, were widely explored, but generally produce poor quality results. Additionally, these approaches require that the time series have exactly the same number of observations. Therefore, approaches involving the measurement of similarity among global characteristics of the time series were subsequently proposed [28]. In our work we present an approach in this direction, with the particularity that we resort to the science of complex networks.

In this chapter we describe the proposed approach and perform cluster analysis in two datasets: a set of simulated time series and a set of real time series. We divide this analysis into two parts: first we analyze the results obtained from the principal components analysis and then we analyze the results obtained from the clustering analysis. Finally we apply the proposed approach to a real dataset and compare the results with those obtained using a traditional approach.

## 5.1 Methods

The approach proposed in this thesis essentially involves the following main tasks:

1. **Generate Complex Networks:** For the sets of time series under analysis we generated five different types of complex networks, namely, NVG, HVG, 10-QG, 50-QG and 100-QG. For the implementation of these networks we use the methods mentioned earlier in the chapter 3.

2. **Calculate Metrics and Normalize:** For each of the complex networks we compute the five topological metrics described in subsection 2.2.2: $\bar{k}$, $\bar{d}$, $S$, $C$, and $Q$. Since the interval of each of the metrics can vary significantly, we apply the Min-Max normalization so that each measure is in the range $[0, 1]$, and we can thus make a better comparison of the metrics and as well as prevent some of them from dominating others in the clustering process.

3. **Dimensionality Reduction:** Use PCA and t - SNE to reduce the dimensionality of the vector of topological measures.

4. **Clustering Analysis:** Finally, we perform the clustering task by feeding the algorithm $k$-means with the previously re-sized feature vectors. We performed the comparison of results obtained using clustering evaluation metrics.

All the tasks are performed in `R` [52] (version 3.4.4) that has a wide set of aid packages, especially an excellent package called `igraph` [18] very useful for graph generation and calculation of metrics.

Before analyzing any dataset we must perform a pre-processing of them, since the data may be incomplete (missing values), noisy (errors or outliers), inconsistent (value discrepancies) and abundant (large volume of data).

Next we explain in more detail two dimensionality reduction techniques that we use to reduce the dimensionality of our dataset that can reach a total of 25 variables for 1700 objects.

### 5.1.1 Dimensionality Reduction

One of the reasons to perform dimensionality reduction is the computational time that the algorithms can take to perform a data mining task. Although this time depends on factors such as: the size of the dataset (number of variables and observations), the task to be performed and the algorithm(s) (iterative or not), in a reduced dataset the performance tends to be faster. Another reason, even more important, is the task of analyzing and understanding the relationship between variables, finding similar cases and outliers.

If we work on a large number of variables (as in the case of this thesis), the ideal is to obtain a representation of the variables in a space of smaller dimensionality that maintains the probability distributions similar to the originals. This reduction is extremely important to obtain a better use of the data, facilitating the extraction of knowledge through a graphical representation and generating a new set of coordinates. On the other hand, we have the notion that when making a reduction in the real space we risk losing information. So we must always have a compromise between reducing the size of the dataset and losing relevant information.

**Principal Component Analysis** Principal Component Analysis (PCA) is the most used dimensionality reduction technique. It is indicated for linearly correlated sets of measures, which can thus be reduced, using an orthogonal transformation (orthogonalization of vectors), to a smaller set of variables, called principal components (PC). The technique allows to transform a set of original inter-correlated variables into a new set of uncorrelated variables (the PCs). The number of PCs is less than or equal to the number of original variables. The goals of PCA are to (a) extract the most important information from the dataset, (b) compress the size of the dataset by keeping only the important information, (c) simplify the description of the dataset, and (d) analyze the structure of the observations and the variables [1].

Suppose that $x$ is a vector of $p$ random variables, and that the variances of the $p$ random variables and the structure of the covariances or correlations between the $p$ variables are of interest. The approach involves look for a few ($\ll p$) derived variables that preserve most of the information given by these variances and correlations or covariances.

The first step is to look for a linear function $\alpha_1' x$ of the elements of $x$ having maximum variance, where $\alpha_1$ is a vector of $p$ constants $\alpha_{11}, \alpha_{12}, \ldots, \alpha_{1p}$ and we transpose, so that:

$$\alpha_1' x = \alpha_{11} x_1 + \alpha_{12} x_2 + \cdots + \alpha_{1p} x_p. \tag{5.1}$$

Next, look for a linear function $\alpha_2' x$ uncorrelated with $\alpha_1' x$ having maximum variance, and so on, so that at the $k$th stage a linear function $\alpha_k' x$ is found that has maximum variance subject to being uncorrelated with $\alpha_1' x, \alpha_2' x, \ldots, \alpha_{k-1}' x$. The $k$th derived variable, $\alpha_k' x$ is the $k$th PC. Up to $p$ PCs could be found, but it is hoped, in general, that most of the variation in $x$ will be accounted for by $m$ PCs, where $m \ll p$ [32].

Let $\Sigma$ be a known covariance matrix of the vector of random variables $x$. The $(i, j)$th element is the (known) covariance between the $i$th and $j$th elements of $x$ when $i \neq j$, and the variance of the $j$th element of $x$ when $i = j$. For $k = \{1, 2, \ldots, p\}$, the $k$th PC is given by $z_k = \alpha_k' x$ where $\alpha_k$ is an eigenvector of $\Sigma$ corresponding to its $k$th largest eigenvalue $\lambda_k$.

We denote by $z$ the vector whose $k$th element is $z_k$, then

$$z = A' x \tag{5.2}$$

where $A$ is the orthogonal matrix whose $k$th column, $\alpha_k$, is the $k$th eigenvector of $\Sigma$. Thus, the PCs are defined by an orthonormal linear transformation of $x$. And

$$\Sigma A = A\Lambda \tag{5.3}$$

where $\Lambda$ is the diagonal matrix whose $k$th diagonal element is $\lambda_k$, the $k$th eigenvalue of $\Sigma$, and $\lambda_k = var(\alpha_k' x) = var(z_k)$.

The eigenvalue associated to a component is equal to the sum of the squared factor scores for this component. Therefore, the importance of an observation for a component can be obtained by the ratio of the squared factor score of this observation by the eigenvalue associated with that component. The value of a contribution is between 0 and 1 and, for a given component, the sum of the contributions of all observations is equal to 1. The larger the value of the contribution, the more the observation contributes to the component [1].

**t-Distributed Stochastic Neighbor Embedding** t-Distributed Stochastic Neighbor Embedding (t-SNE) is a relatively recent nonlinear dimensionality reduction technique, particularly suitable for visualization of high-dimensional datasets that produces substantially better results than alternative techniques. It is based on the idea that an efficient visualization of high-dimensional data is obtained by representing each data object as a two-dimensional (or at most three-dimensional) point such that similar objects are represented by nearby points and different objects are represented by distant points with high probability. The resulting two-dimensional points can be viewed on a scatter plot. This leads to a map of the data that reveals the underlying structure of objects, such as the presence of clusters.

t-SNE minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding [62]. Assume a dataset (high-dimensional) of objects $\{x_1, x_2, \ldots, x_n\}$ and a function that computes the distance between pairs of objects, for example, the Euclidean distance. The aim is to learn an $s$-dimensional embedding in which each object is represented by a point $\{y_i, y_2, \ldots, y_n\}$ with $y_i \in \mathbb{R}^s$ (typically $s = \{2, 3\}$). To this, t-SNE defines joint probabilities $p_{ij}$ that measure the pairwise similarity between objects $x_i$ and $x_j$ by computing two conditional probabilities:

$$p_{j|i} = \frac{exp(-d(x_i, x_j)^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-d(x_i, x_k)^2/2\sigma_i^2)}, p_{i|i} = 0,$$
$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{5.4}$$

the bandwidth of the Gaussian kernels, $\sigma_i$, is set in such a way that the perplexity of the conditional distribution $P_i$ equals a predefined perplexity $u$. As a result, the optimal value of $\sigma_i$ varies per object: in regions of the data space with a higher data density, $\sigma_i$ tends to be smaller than in regions of the data space with lower density. In the $s$-dimensional embedding, the similarities between two points $y_i$ and $y_j$ (that is, the low-dimensional models of $x_i$ and $x_j$) are measured using a normalized heavy-tailed kernel. Specifically, the embedding similarity $q_{ij}$

between the two points $y_i$ and $y_j$ is computed as a normalized Student-t kernel with a single degree of freedom. The heavy tails of the normalized Student-t kernel allow dissimilar input objects $x_i$ and $x_j$ to be modeled by low-dimensional counterparts $y_i$ and $y_j$ that are too far apart. This is desirable because it creates more space to accurately model the small pairwise distances in the low-dimensional embedding. The locations of the embedding points $y_i$ are determined by minimizing the Kullback-Leibler divergence between the two distributions in relation to the locations of the points on the map. Due to the asymmetry of the Kullback-Leibler divergence, the objective function focuses on modeling high values of $p_{ij}$ by high values of $q_{ij}$ [62].

### 5.1.2 $k$-means Algorithm

There are two large groups of clustering algorithms, *hierarchical* and *partial*. Hierarchical clustering algorithms recursively encounter nested clusters in agglomerative mode (starting with each data point in its own cluster and merging the most similar pair of clusters in succession to form a cluster hierarchy) or in divisive mode (starting with all data points in a cluster and recursively dividing each cluster into smaller clusters). In comparison to hierarchical clustering algorithms, partial clustering algorithms locate all clusters simultaneously as a data partition and do not impose a hierarchical structure [31], divide the dataset into several mutually exclusive clusters, and then maximize the similarity intra-clusters and minimize similarity inter-clusters, moving objects from one cluster to another.

In clustering analysis there are several algorithms that can be applied to the set of characteristics used for our analysis. We decided to use the *k-means* (a partial algorithm) algorithm that is the most commonly used, is fast and produces good results. However has the disadvantage of needing the specification of an *a priori* number of clusters. But since we only want to show the power of the proposed approach and know the number of classes in each set of datasets used, this is not a problem at this stage, and also we can always estimate the best number of clusters to use.

$k$-means receives an array of patterns $n \times d$, where $n$ objects are embedded in a $d$-dimensional resource space. The algorithm locates a partition in such a way that the square error between the empirical average of a cluster and the points in the cluster is minimized, that is, it minimizes the sum of the square error on all $k$ clusters.

The main steps of the k-means algorithm are:

1. Randomly generated, within the data domain, $k$ initial "means" corresponding to $k$ clusters,
2. $k$ clusters are formed by associating every observation with the nearest mean,
3. the centroid of each of the $k$ clusters formed become the new mean,
4. repeat steps 2 and 3 until the clusters are stable.

### 5.1.3 Cluster Evaluation

In order to evaluate the best results of the different cluster analyzes we will use essentially a metrics evaluation clustering called *adjusted Rand index* and the *average silhouette* to evaluate the quality of the clusters formed. The two measures have different functions, the first is a measurement of the accuracy of the results: compares the clusters obtained with the true clusters. The second measures the quality of clusters obtained without knowledge of the true clusters.

**Adjusted Rand Index** *Rand Index* is a measure of similarity between two data clusters. It is related to **precision**, but it is applicable even when class labels are not used. By definition, given a set $S$ of $n$ elements and two partitions from $S$ to compare, $X$ a partition from $S$ in $r$ partitions and $Y$ a partition from $S$ in $s$ partitions, the Rand index is given by [12]:

$$\omega = \frac{a+b}{a+b+c+d} \tag{5.5}$$

where:

- $a$ represents the number of pairs of elements in $S$ that are in the same subset $X$ and in the same subset $Y$,
- $b$ represents the number of pairs of elements in $S$ that are in different subsets $X$ and in different subsets $Y$,
- $c$ represents the number of pairs of elements in $S$ that are in the same subset $X$ and in different subsets $Y$, and
- $d$ represents the number of pairs of elements in $S$ that are in different subsets $X$ and in the same subset $Y$.

The Rand Index has a value between 0 and 1, where 0 indicates that the two clusters of data do not agree on any pair of points and 1 indicates that the data groupings are exactly the same.

One of the main criticisms against the original Rand Index is that it is not corrected to chance, that is, the expected value of the Rand Index between two random partitions is not a constant. The Adjusted Rand Index assumes the generalized hyper-geometric distribution as the randomness model. By definition, given a set $S$ of $n$ elements and two partitions from $S$ to compare, $X$ a partition from $S$ in $r$ partitions and $Y$ a partition from $S$ in $s$ partitions, the Adjusted Rand Index is given by [12]:

$$\omega_A = \frac{a - \frac{(a+c)(a+b)}{d}}{\frac{(a+c)+(a+b)}{2} - \frac{(a+c)(a+b)}{d}} \tag{5.6}$$

which is corrected for chance under the assumption that the number of groups (classes/clusters) in both the $X$ and $Y$ partitions to be compared is the same. The Adjusted Rand Index takes values between $-1$ and 1. It is negative if the index is less than the expected index. Its expected value is 0 in the case of random clusters. A larger Adjusted Rand Index means a higher agreement between two partitions.

**Average Silhouette** *Silhouette* is a measure that represents how well each object is within its cluster, that is, how similar it is to its own cluster (**cohesion**) compared to other clusters (**separation**).

Formally, for each object $i$, let $a(i)$ be the average distance between $i$ and all other objects within its cluster (the lower the better), and $b(i)$ the shortest average distance between $i$ for all objects in any other cluster, that $i$ does not belong. The cluster with the lowest mean dissimilarity is said to be the *neighbor cluster* of $i$. Silhouette is defined as:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{5.7}$$

The silhouette can be calculated with any distance metric. Its value ranges from $-1$ to 1, where a high value indicates that the object is well compatible with its own cluster but not with neighbor clusters. If most objects have a high value, the cluster configuration is appropriate. In this work , we use the Euclidean distance.

The average silhouette value is given by the arithmetic mean of the silhouette value for all objects.

## 5.2 Clustering Time Series Models

The results in chapter 4 indicate the different time series models stand out in at least one topological metric of the several networks, evidencing the possibility of time series classification by unsupervised learning. In this section we present the main results of this classification. For this we use the set of simulated in chapter 4. Recalling, our main dataset consists of a total of 1700 time series of length 10000 simulated from 17 different models with 100 replicates. For a more rigorous analysis we simulate a similar set in which each series has a significantly smaller length, 1000 observations, to show that our method is advantageous for both large time series and small time series. To the first set we will refer as set $n_1$ and the second as $n_2$.

In order to understand which mappings contribute to a better grouping of the series we decided to use 31 different feature vectors in the clustering analysis, corresponding to all possible combinations of the five mappings. The objective of this exhaustive study is to assess whether the use of two different concepts of mapping methods can be a significant advantage for a better recognition of the time series characteristics through complex networks.

Next we present these results based on different types of dimensionality reduction techniques.

**First Experience** In the first analysis we use PCA as a dimensionality reduction technique and *a priori* knowledge of the correct number of clusters in the dataset, that is, $k = 17$.

The results of the clustering evaluation metrics obtained for the different combinations of feature vectors for each of the sets $n_1$ and $n_2$ are presented in the table 5.1, columns 2 and 3, and columns 4 and 5, respectively. The colors represent the three maximum values of the corresponding column, with the darker color highlighting the maximum value and the lighter color the third maximum value.

From the analysis of table 5.1 we see that the feature vector that is closest to the real clusters is the one corresponding to HVGs, 50-QGs and 100-QGs, with a value of 0.63 in a range of $[-1, 1]$, in the case of the set $n_1$, and the vectors corresponding to 10-QGs and 50-QGs or to 10-QGs and 100-QGs or NVGs, 10-QGs and 50-QGs, which obtained a value of 0.46, in the case of the set $n_2$. While the vector that obtained the best value of the average silhouette was the corresponding 10-QGs, which also obtained a fairly good value of adjusted Rand index (0.59), in the case of the set $n_1$, and the vector corresponding to the 10-QGs and 50-QGs, in the case of the set $n_2$.

We can also verify that, using only one mapping method, the methods based on transition probability are much better than the methods of visibility, which is in agreement with the expected one, since this method better captures the variability of the observations of the time series. And they are also among all sets that result in better quality clusters.

The addition of more information in some cases translates into a better result, as is the case of the use the 50-QGS metrics that always obtain better results when we join other metrics of other types of graphs, as in the case of metrics of NVGs. However, as expected, the addition of more information about the data does not always translate into a better result, for example in the case of the use metrics of QGs of 10 nodes, the addition of other types of graphs always decreases the results. This is no longer true for the set $n_2$ the results are mostly better when we use the 10-QGs in conjunction with other types of graphs.

In summary, the results are relatively better when we apply our approach to sets of longer time series, which was already an expected result given that a larger set of observations can provide greater power to characterize the series, and complex networks are really powerful for analyzing large datasets. Transition probability methods are better at characterizing the series than the visibility methods, but when visibility methods are coupled with transition probability methods this improves the ability to characterize the temporal series.

The best average silhouette values do not coincide with the best values of the adjusted Rand index. That is understandable since the two measure have different functions, the first measures the quality of clusters obtained that has no knowledge of the true results, and the second is a measurement accuracy of the results, that is, it compares the clusters obtained with the true clusters.

| Mappings | $n_1 = 10000$ | | $n_2 = 1000$ | |
|---|---|---|---|---|
| | **Adjusted Rand Index** | **Average Silhouette** | **Adjusted Rand Index** | **Average Silhouette** |
| **NVG** | 0.19 | 0.34 | 0.13 | 0.34 |
| **HVG** | 0.40 | 0.49 | 0.23 | 0.40 |
| **Q10** | **0.59** | **0.66** | **0.39** | **0.50** |
| **Q50** | **0.43** | **0.60** | 0.43 | 0.45 |
| **Q100** | **0.56** | **0.62** | 0.33 | 0.42 |
| **NVG-HVG** | 0.46 | 0.44 | 0.32 | 0.38 |
| **NVG-Q10** | 0.51 | 0.56 | 0.40 | 0.36 |
| **NVG-Q50** | 0.49 | 0.48 | **0.44** | **0.44** |
| **NVG-Q100** | 0.53 | 0.53 | 0.36 | 0.41 |
| **HVG-Q10** | 0.48 | 0.54 | 0.42 | 0.43 |
| **HVG-Q50** | 0.46 | 0.48 | 0.40 | 0.40 |
| **HVG-Q100** | 0.55 | 0.52 | 0.39 | 0.42 |
| **Q10-Q50** | 0.52 | 0.58 | **0.46** | **0.51** |
| **Q10-Q100** | 0.53 | 0.58 | **0.46** | **0.49** |
| **Q50-Q100** | 0.53 | 0.58 | 0.36 | 0.45 |
| **NVG-HVG-Q10** | 0.40 | 0.57 | 0.41 | 0.38 |
| **NVG-HVG-Q50** | **0.59** | **0.48** | 0.39 | 0.38 |
| **NVG-HVG-Q100** | **0.61** | **0.51** | 0.35 | 0.37 |
| **NVG-Q10-Q50** | 0.53 | 0.52 | **0.46** | **0.46** |
| **NVG-Q10-Q100** | 0.52 | 0.52 | **0.44** | **0.43** |
| **NVG-Q50-Q100** | 0.55 | 0.53 | **0.44** | **0.44** |
| **HVG-Q10-Q50** | 0.50 | 0.52 | 0.37 | 0.41 |
| **HVG-Q10-Q100** | 0.53 | 0.50 | 0.39 | 0.44 |
| **HVG-Q50-Q100** | **0.63** | **0.57** | **0.45** | **0.45** |
| **Q10-Q50-Q100** | 0.51 | 0.56 | **0.44** | **0.47** |
| **NVG-HVG-Q10-Q50** | 0.50 | 0.52 | 0.41 | 0.43 |
| **NVG-HVG-Q10-Q100** | 0.53 | 0.50 | 0.39 | 0.42 |
| **NVG-HVG-Q50-Q100** | 0.56 | 0.55 | 0.42 | 0.40 |
| **NVG-Q10-Q50-Q100** | 0.50 | 0.52 | **0.45** | **0.42** |
| **HVG-Q10-Q50-Q100** | 0.52 | 0.55 | **0.45** | **0.46** |
| **NVG-HVG-Q10-Q50-Q100** | 0.52 | 0.53 | 0.42 | 0.43 |

Table 5.1: Clustering evaluation metrics for the different clustering analysis performed using PCA as a dimensionality reduction technique. The results refer to the evaluation metrics for the dataset with time series of length 10000, columns 2 and 3 and length 1000,columns 4 and 5.

**Second Experience** A second analysis involved the addition of the t-SNE technique to the previous approach in order to obtain a better separation of the clusters by increasing the results of the evaluation metrics. Also here we indicate *a priori* the correct number of clusters of the dataset.

The results obtained for each of the experiments using the different combinations of feature vectors for each of the sets $n_1$ and $n_2$ are presented in the table 5.2.

The results in table 5.2 (adjusted Rand index and average silhouette) are quite different from the results in table 5.1, specially for the set $n_2$.

Although the previous analysis showed no great advantage in using the graphs based on visibility together with the QGs, these new results, based on a different technique for dimensionality reduction that aims at a better distribution of the observations by the different groups, indicate an increase of the precision in the clustering.

The combination of the concepts of mappings of visibility and probability of transitions can capture in some cases more information of the time series than that captured using only one of the concepts. A possible explanation is that visibility is able to capture essentially the structural properties of the time series and transitions probabilities manage to capture the properties of the data variability.

**Third Experience** A final test involves determining the best value of $k$, based on the adjusted Rand index.

A summary of these results are presented in the tables D.1 and D.2 in the appendix D. We can conclude that the results are similar to those obtained previously since the values of $k$ are mostly between 15, 16 and 17. Note that for $n_2$ using PCA the values of $k$ do vary more.

For the first experience (5.2), the highest adjusted Rand index value was obtained for $k = 15$, and for the second test for $k = 17$. This reinforces once again the advantage of using t-SNE in the clustering task.

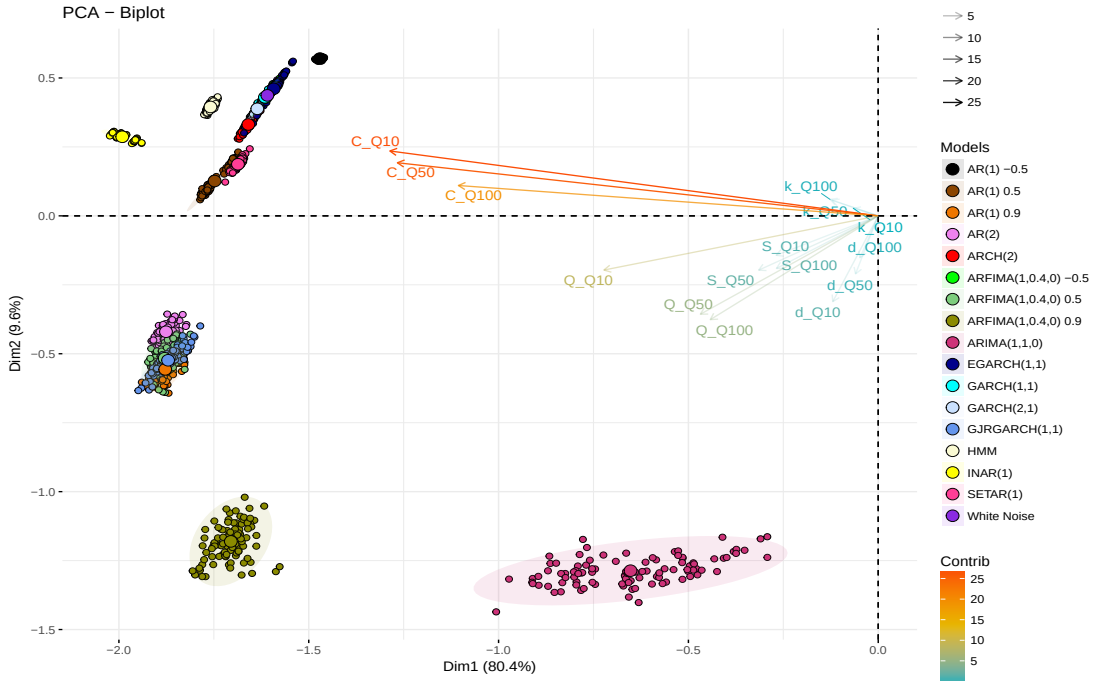
We will now discuss some of the results obtained. Let us focus on two vectors of specific features: one containing the topological metrics of the NVGs, HVGs, 50-QGs and 100-QGs (best result of the adjusted Rand index obtained in the second test for the set $n_1$) and one containing the topological metrics of the 10-QGs, 50-QGs and 100-QGs (best average silhouette result obtained in the second test for the set $n_1$). We will divide this analysis into two parts: the first part on the results obtained by the PCA and the second part on the resulting clusters.

| Mappings | $n_1 = 10000$ | | $n_2 = 1000$ | |
|---|---|---|---|---|
| | **Adjusted Rand Index** | **Average Silhouette** | **Adjusted Rand Index** | **Average Silhouette** |
| **NVG** | 0.25 | 0.49 | 0.21 | 0.45 |
| **HVG** | 0.57 | 0.60 | 0.35 | 0.54 |
| **Q10** | 0.61 | 0.66 | **0.51** | **0.62** |
| **Q50** | **0.52** | **0.68** | 0.53 | 0.59 |
| **Q100** | **0.56** | **0.70** | 0.40 | 0.59 |
| **NVG-HVG** | 0.54 | 0.58 | 0.38 | 0.53 |
| **NVG-Q10** | 0.62 | 0.63 | 0.46 | 0.52 |
| **NVG-Q50** | 0.59 | 0.64 | 0.57 | 0.58 |
| **NVG-Q100** | **0.66** | **0.67** | 0.50 | 0.53 |
| **HVG-Q10** | 0.61 | 0.65 | 0.52 | 0.59 |
| **HVG-Q50** | 0.59 | 0.63 | **0.60** | **0.59** |
| **HVG-Q100** | 0.63 | 0.66 | **0.55** | **0.62** |
| **Q10-Q50** | **0.64** | **0.67** | **0.55** | **0.61** |
| **Q10-Q100** | 0.57 | 0.65 | **0.57** | **0.61** |
| **Q50-Q100** | 0.65 | 0.67 | 0.46 | 0.58 |
| **NVG-HVG-Q10** | 0.61 | 0.63 | 0.53 | 0.53 |
| **NVG-HVG-Q50** | 0.67 | 0.63 | 0.50 | 0.56 |
| **NVG-HVG-Q100** | **0.69** | **0.65** | 0.48 | 0.54 |
| **NVG-Q10-Q50** | 0.63 | 0.63 | **0.58** | **0.57** |
| **NVG-Q10-Q100** | 0.61 | 0.63 | 0.54 | 0.58 |
| **NVG-Q50-Q100** | **0.71** | **0.66** | **0.56** | **0.60** |
| **HVG-Q10-Q50** | 0.63 | 0.61 | **0.56** | **0.60** |
| **HVG-Q10-Q100** | 0.59 | 0.61 | 0.53 | 0.56 |
| **HVG-Q50-Q100** | 0.65 | 0.65 | **0.59** | **0.61** |
| **Q10-Q50-Q100** | **0.63** | **0.70** | 0.54 | 0.60 |
| **NVG-HVG-Q10-Q50** | 0.65 | 0.62 | 0.54 | 0.57 |
| **NVG-HVG-Q10-Q100** | 0.67 | 0.61 | 0.52 | 0.54 |
| **NVG-HVG-Q50-Q100** | **0.72** | **0.68** | 0.53 | 0.54 |
| **NVG-Q10-Q50-Q100** | 0.64 | 0.66 | **0.60** | **0.58** |
| **HVG-Q10-Q50-Q100** | 0.64 | 0.62 | **0.59** | **0.59** |
| **NVG-HVG-Q10-Q50-Q100** | 0.66 | 0.62 | 0.55 | 0.57 |

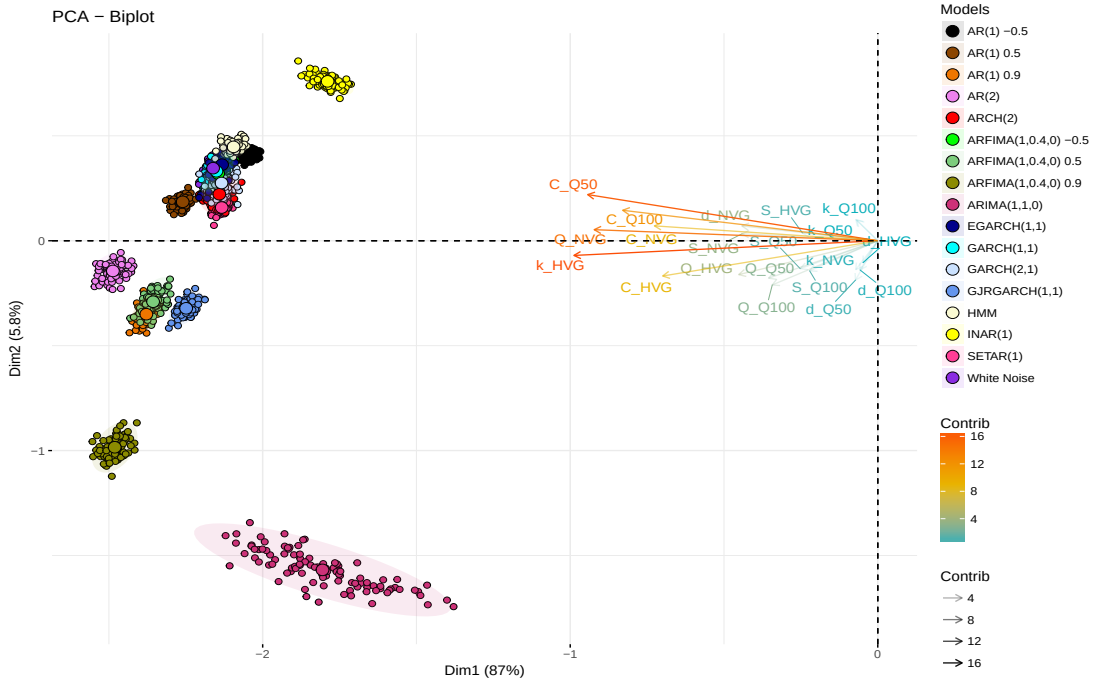
Table 5.2: Presentation of the clustering evaluation metrics for the different clustering analysis performed through different combinations of mappings and using t-SNE as a dimensionality reduction technique. The results refer to the evaluation metrics for the dataset with time series of length 10000 in columns 2 and 3, and of length 1000 in columns 4 and 5.

### 5.2.1 PCA Results

Figure 5.1 represents two biplots obtained by the PCA, one for the best adjusted Rand index obtained in the second experiment and another for the best average silhouette. It should be recalled that the results analyzed here have not yet suffered the reduction imposed by t-SNE.




(a) Q10-Q50-Q100




(b) NVG-HVG-Q50-Q100

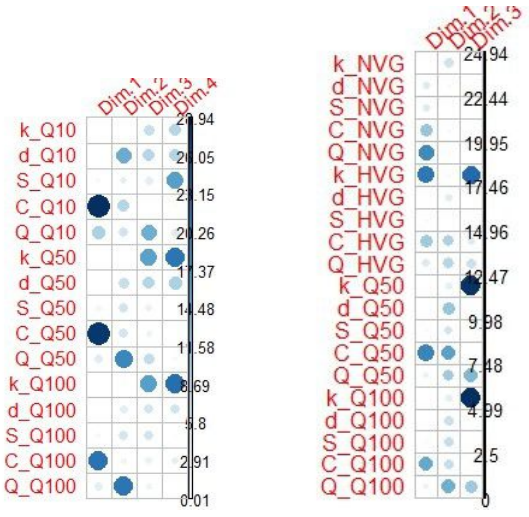
Figure 5.1: Results of the PCA analysis. Objects belonging to different groups have different colors, and the arrows represent the contributions of the features to the PCs (the larger the size, sharpness, and closer to orange the greater the contribution of the feature).

We can start by noting that several of the objects belonging to different classes are actually separated into clusters in this bidimensional space. We can also verify that the apparently more dissimilar objects are those corresponding to models with clearer trends (ARIMA and ARFIMA0.9) and the counting models (INAR), which were also the ones highlighted in section 4.1.

If we analyze the plots separately we see some differences. In the plot 5.1b we find groups corresponding to the periodic models AR(2), the GJR-GARCH and the AR(1)0.9 together with the ARFIMA0.5 models, which in plot 5.1a are overlapping. These results are also in agreement with the characterization that we have done previously, that in several topological measures the values for the AR(1)0.9 networks are very similar to the ARFIMA0.5 networks. However the addition of a new mapping method approximated the results for the HMM networks to the results of the AR(1)-0.5 networks and the remaining ones, which in the plot 5.1a were apparently more separated. The same happens for the AR(1)0.5 and SETAR networks.

This really shows the advantages (and disadvantages) of embedding more information. However, we can still verify that, just as the arrows in the plots themselves suggest, the 50-QGs and 100-QGs contribute to distinguish the INAR, ARIMA, and ARFIMA0.9 networks, the HVGs contribute to distinguish AR(2) and GJR-GARCH networks, the 100-QGs for the AR(1)0.5 and SETAR networks, and finally the NVGs seem to contribute to a greater disposition of the values corresponding to the fairly aggregated groups in the second quadrant.

To better analyze which specific metrics contribute the most to each PC, we look at the figure 5.2. We see that the two sets of PCs are different. In the left panel, quantiles only mappings, (5.2a) four PCs are required to explain 95% (the value we choose) of the variability in the data, while in the right panel 5.2b only three PCs are required.




(a) Q10-Q50-Q100 (b) NVG-HVG-Q50-Q100

Figure 5.2: Correlation plot to highlight the features that contribute most for each of the first few principal components.

In the first case, clustering coefficient of the QGs are those that contribute most to CP1 (which represents 80.4% of the data), and the average path length of 10-QGs and the modularity of the 50-QGs and 100-QGs contribute to CP2 (which represents 9.6% of the data). Other metrics such as average degre and number of communities contribute to the remaining CPs.

In the second case, the modularity of NVGs, average degres of the HVGs and clustering coefficient of the 50-QGs contribute to the CP1 (which represents 87% of the data), and the clustering coefficient of 50-QGs and modularity of the 100-QGs for CP2 (which represents 5.8% of the data). The CP4 is heavily contributed by the average degree of HVGs and QGs.

### 5.2.2 Clusters Results

In figure 5.3b we can note the almost perfect assignment of the objects by the different clusters. The exceptions are `ARFIMA-0.5`, `EGARCH`, `GARCH1`, `GARCH2` and `WN` networks which are not distinguishable and are assigned to the same clusters.

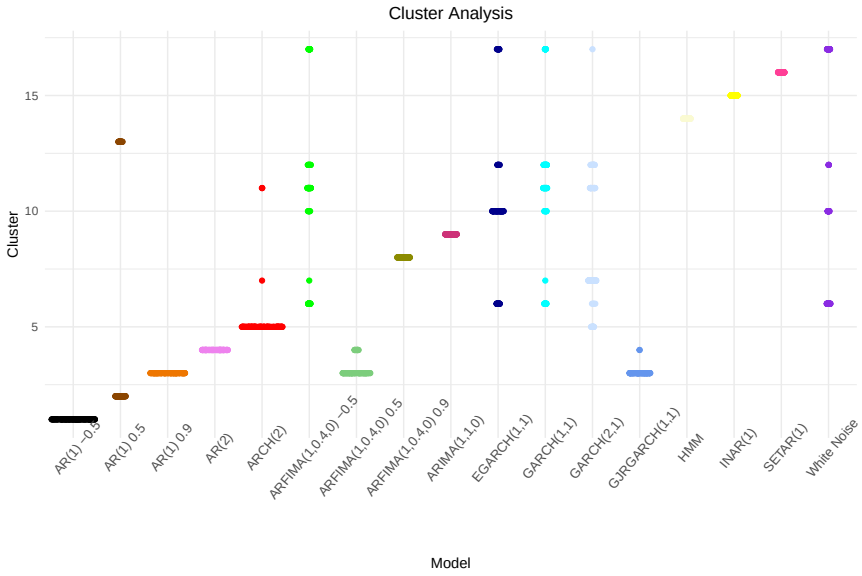
When only the features of the QGs are used, the `AR(1)0.9`, `ARFIMA0.5` and `GJR-GARCH` networks are assigned to the same cluster, thus contributing to the lower result of the adjusted Rand index.

We conclude that the most similar models from the point of view of the complex networks, and consequently more difficult to distinguish, are the models of the `GARCH` family (except the `GJR-GARCH` that even in the representation of time series are easily distinguishable) and `WN` that in the perspective of the analysis of time series have very similar characteristics. To these is added the `ARFIMA` models with negative coefficient of regression.

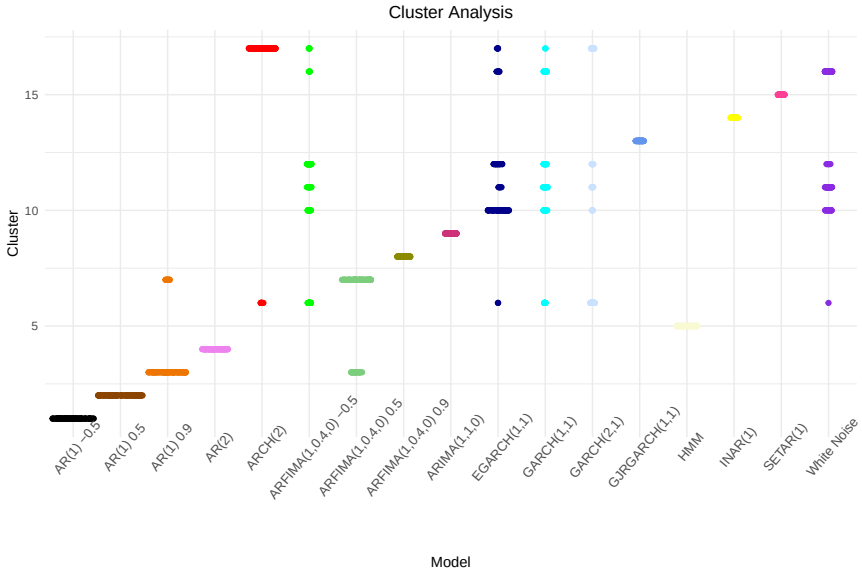
Recalling, silhouette analysis intuitively is used to measure the space between the resulting clusters. The silhouette plot displays a measure of how each object in a cluster is close to objects in neighboring clusters. This measure has a range of $[-1, 1]$.

The silhouette coefficients close to 1 indicate that the object is far from the neighboring clusters. A 0 value indicates that the object is very close to the decision limit between two neighboring clusters, and a negative value indicates that the object was probably assigned to the wrong cluster. As we can see in figure 5.4, there are some negative values. In the first plot only objects assigned to cluster 5 were probably wrongly assigned, however these values are more significant than those presented in the second plot that presents negative values in the cluster 6, 11 and 12. These values make sense because they correspond to clusters (see corresponding colors in figure 5.3) that do not contain just the correct objects.

The width of the bars represent the size of the cluster and so, ideally all the bars should have the same width. In 5.4a the cluster 3 has a much larger width than the others (since it contains objects of type `AR(1)0.9` and of type `ARFIMA0.5`) and, in contrast, clusters 7, 12 and 13 have a width which is also not desirable. On the other hand, in figure 5.4b all the plots are more or less similar indicating that there is a good structure for the clusters, with most of the



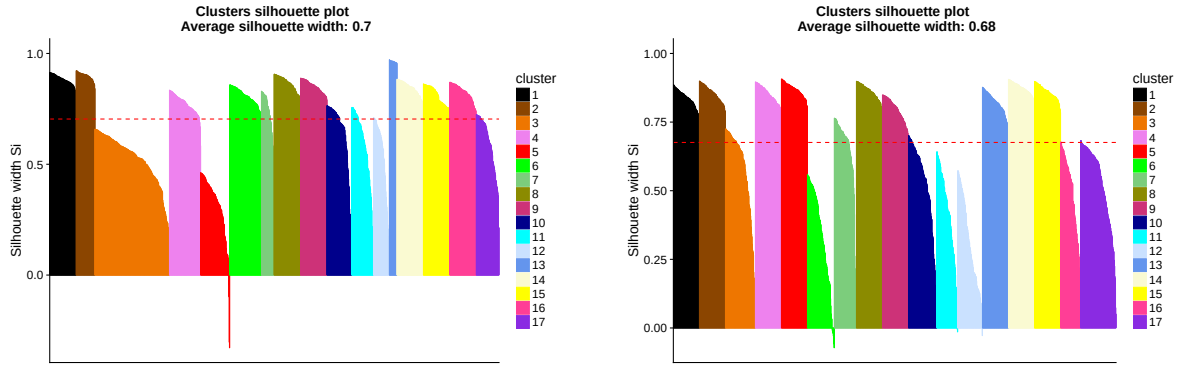

(a) Q10-Q50-Q100




(b) NVG-HVG-Q50-Q100

Figure 5.3: Plot the distribution of the objects corresponding to the time series models by the different clusters.

observations seemed to belong to the cluster in which they are. Although the value of average silhouette is slightly lower than the first.




(a) Q10-Q50-Q100

(b) NVG-HVG-Q50-Q100

Figure 5.4: Silhouette scores from seventeen types of complex networks from time series models.

## 5.3 Clustering *Synthetic Control* Time Series

As we mentioned in section 4.2 the distinction between different types of patterns is essential to automatically detect potential failures in the process being monitored and decide on the solution of the problem depending on the type of pattern involved. Let us now assess the approach proposed for the unsupervised classification of the *synthetic control* time series dataset.

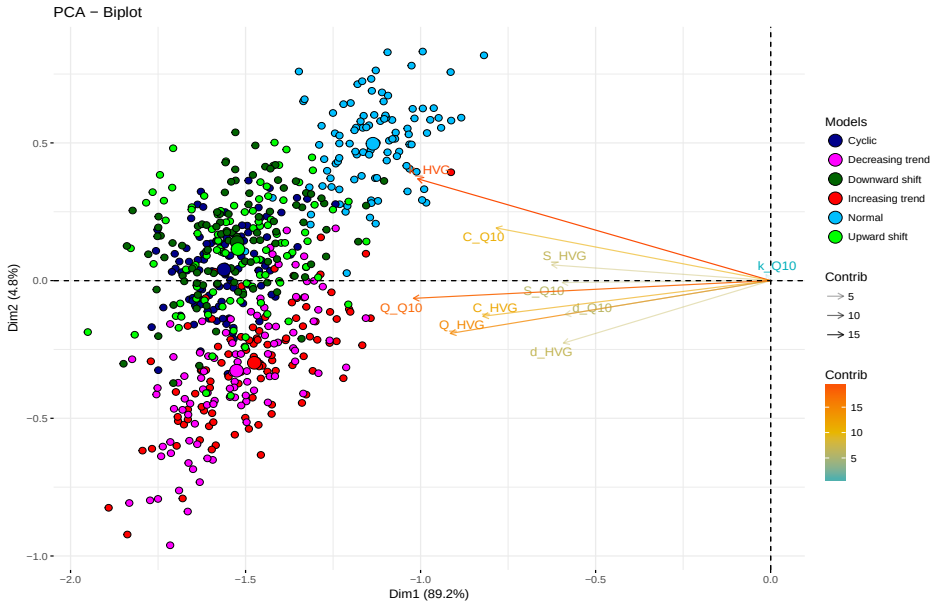
For this particular dataset we will make a small change in the tasks mentioned in section 5.1 since they are rather small series ($T = 60$). The change affects the task 1 where we will only generate the complex networks corresponding to the HVG and 10-QG mappings since they were the ones that showed to be more useful to distinguish the characteristics of the series in the previous chapter.

Next, we present the PCA results obtained from the features vector of these mappings and the results of the clusters obtained after the application of t-SNE.

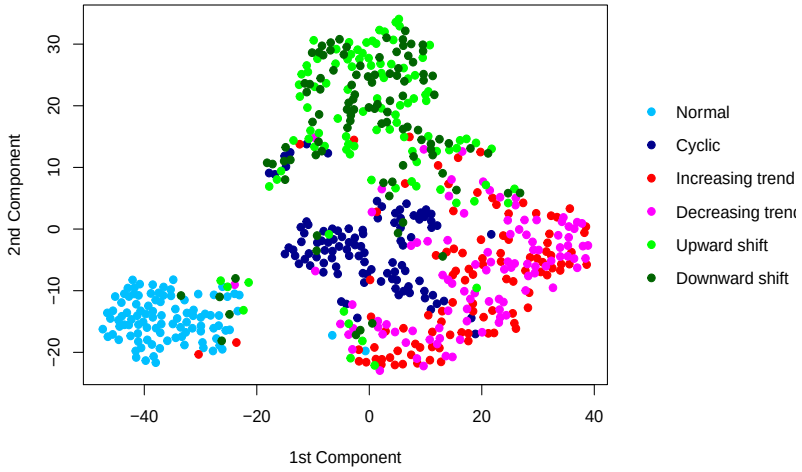
### 5.3.1 PCA Results

The figure 5.5 illustrates two biplots related to the data dimensionality reduction task, the upper panel illustrates the results obtained by the PCA, where the contribution of the network metrics can be analyzed, and the lower panel illustrates the results obtained by t-SNE after PCA.

The results indicate that the HVGs are especially good at capturing the special trends properties of time series, using average path length ($\bar{d}$), modularity ($Q$) and clustering coefficient ($C$). This is in line with our statements in section 4.2. Just as the average degree ($\bar{k}$) is quite important to characterize the normal patterns.




(a) PCA




(b) t-SNE

Figure 5.5: (a) results of the PCA analysis, the arrows represent the contributions of the features to the principal components (the larger the size, sharpness, and closer to the orange color the greater the contribution of the feature to the principal component); (b) results of t-SNE analysis after PCA.

On the other hand, the clustering coefficient ($C$) results obtained for the 10-QGs appear to be good at distinguishing normal and shift patterns from the others, and the number of communities ($S$) very good to distinguish shift and cyclic patterns. The modularity ($Q$) contributes to the distinction between the shift and trend patterns of others.

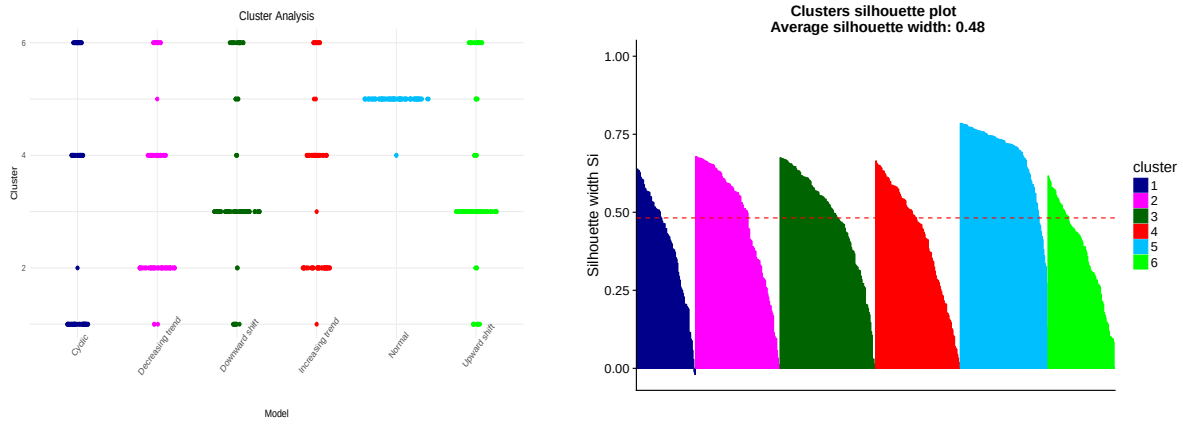
In the characterization performed in the previous chapter no topological measure completely distinguished the cyclic pattern. Here, this result also subsists. However, the application of t-SNE allows a better separation of the data in the different main types of time series patterns: normal, trend, shift and cycle. This separation is not perfect since cyclic patterns are still between trend and shift patterns, due to the metrics of the 10-QGs. Also and as expected, it is not possible to distinguish neither between the two different shift patterns the two different trend patterns.

The metrics of these networks are thus promising to distinguish the patterns.

### 5.3.2 Clusters Results

In order to perform the cluster analysis of the *synthetic control* time series we will assign to the $k$-means algorithm the vector of features reduced by the PCA and t-SNE and let us indicate *a priori* the number of true clusters, that is, $k = 6$.

Figure 5.6a shows many objects attributed to wrong clusters. However, this was already expected for all that we have mentioned above and in the section 4.2. Hence the value obtained for the metric evaluation adjusted Rand index is 0.37. The clusters formed are relatively good, if we analyze the figure 5.6b we can verify that the thicknesses of the bars are relatively similar, and the bars corresponding to the clusters 2, 3 and 5 are those that have more scores above the average silhouette (0.48), mainly the cluster 5. These clusters represent the trend, shift and normal patterns, respectively (see figure 5.6a together).




(a) Distribution of the objects by the clusters

(b) Clusters silhouette plot

Figure 5.6: Two plots showing (a) silhouette scores and (b) distribution of the objects corresponding to the time series patterns by the different clusters.

As previously discussed the cyclic patterns are more difficult to distinguish from the other and here this is reflected in the thickness of the thinner bar and in the presence of some negative scores.

## 5.4 Clustering Real Time Series

In order to show the potential of the proposed approach we apply our approach to a set of time series datasets and compare it with the characteristic-based clustering for time series proposed by [64], which we call the "traditional approach". First we present the basic idea of this approach and analyze the results obtained.

**Traditional Approach** The "traditional approach" is also based on the use of statistical measures as identifying characteristics. It consists of the use of a set of thirteen measures extracted from the time series, namely classic statistical measures such as trend, seasonality, periodic (frequency), serial correlation, skewness, and kurtosis, and advanced special measures such as nonlinearity structure, self-similarity, and chaos. Four of the previous measures (serial correlation, non-linearity, skewness, and kurtosis) are calculated in both the raw time series as well as in the de-trended and de-seasonalized time series [64].

To apply the "traditional approach" we calculate the thirteen measures using the functions in [29], optimized versions of [64].

**Data** The real data consists of a set of nine real time series datasets widely used in classification tasks. One of the datasets, "18Pairs", was extracted from the package `TSclust` [44] of `R` and represents a set of data formed by pairs of time series of different domains. The remaining sets were extracted from the UCR Time Series Classification Archive [13]. We have taken care to choose, within a vast set of hypotheses, time series representing different systems, from extracted series from ECGs, to motion through accelerometers, to electrical devices and sound sensors. These series are of general interest and interpretable. In table 5.3 we present more details about each of these sets, namely the number of time series in each set, the size of the series, and the number of classes in each set of series.

We also want to explain that if we analyze the number of time series of each set under analysis presented in [13], this value in some of the sets, will be slightly higher than the corresponding value presented in table 5.3. This is because the "traditional approach" involves parametric methods, in particular, the method used to find the period of a time series uses an estimate of the spectral density of the series by fitting an AR model (autoregressive model) that involves an arbitrary threshold chosen by trial and error [29]. This process translates into a problem when applied to some of the time series under analysis, since the frequency found for such series is greater than its own length which is clearly not correct. A possible solution would be to increase the threshold that was defined by trial and error, however, given that the number of time series in which this error happens is quite small when compared to the amount of time series present in the set, we decided to discard those series.

| Dataset | Size of Set | Time Series Length | Num. of Classes | Adj. Rand index $[-1, 1]$ | | Avg. Silhouette $[-1, 1]$ | |
|---|---|---|---|---|---|---|---|
| | | | | Traditional approach | Proposed approach | Traditional approach | Proposed approach |
| **18Pairs** | 36 | 1000 | 18 | 0.17 | **0.67** | 0.18 | **0.32** |
| **CinC_ECG_torso** | 1420 | 1639 | 4 | 0.06 | **0.38** | 0.49 | **0.51** |
| **Cricket_X** | 773 | 300 | 12 | **0.14** | 0.05 | **0.50** | 0.41 |
| **ECG5000** | 4987 | 140 | 5 | **0.05** | 0.01 | 0.39 | **0.42** |
| **ElectricDevices** | 16631 | 96 | 7 | **0.30** | 0.22 | 0.34 | **0.35** |
| **FordA** | 4921 | 500 | 2 | **0.06** | 0.00 | 0.36 | **0.42** |
| **FordB** | 4446 | 500 | 2 | **0.18** | 0.02 | 0.38 | **0.39** |
| **InsectWingbeatSound** | 2196 | 256 | 11 | 0.06 | **0.15** | 0.45 | 0.45 |
| **UWaveGestureLibAll** | 4457 | 945 | 8 | 0.08 | **0.20** | **0.49** | 0.42 |

Table 5.3: Brief description of the time series datasets analyzed; clustering evaluation metric (adjusted Rand index and average silhouette), obtained for the "traditional approach" and for the proposed approach. The values in bold represent the best results. Shaded values highlight the values that cause a difference with the value compared to greater than 0.10.

### 5.4.1 Clustering Results

Again, given the diversity of possible mappings we can use in the clustering task (as sees in section 5.2), we will make a small change to the tasks mentioned in 5.1. As the sets analyzed here are relatively different, that is, they have different quantities of time series in each set, the lengths of the time series are also variable (between $T = 96$ and $T = 1639$) and the number of classes also varies, we decided to use a set of mappings in particular, the NVG, HVG and 50-QGs, since they were the ones that showed more promising results after some tests of different combinations of all the mappings.

After generating all the data needed to feed the $k$-means algorithm, we started the clustering analysis for each of the sets of time series separately. We summarize the values of the evaluation metrics corresponding to each set in table 5.3.

The results indicate that the proposed approach yields better results in the average silhouette for almost all datasets. As for the adjusted Rand index, we only managed to improve in 4 out of 9 of the sets. However, in the cases where we are better, there is a much larger gain when compared to the traditional approach (namely in the first two sets, where the value differences are 0.50 and 0.32), while in the cases were are surpassed the margin is much smaller.

Looking at the lengths, it seems that our method benefits from longer time series, with the largest gain in the adjusted Rand index coming precisely from the three longest time series (with $T = 1639$, $T = 1000$ and $T = 945$). This concurs with what we have observed before in chapter 4, where time series with more data points gave origin to more spread out values

that could potentially better separate between classes. Further tests would need to be made to confirm this intuition and more formally prove that this is indeed the case.

Considering that our approach is based on a novel concept and that there is still a wide range of topological metrics to explore, we consider these preliminary results show that our approach can at the very least become quite competitive.

The main advantage of our proposed methodology is that it is a completely nonparametric method that can serve as an alternative to the parametric methods of time series analysis.

# Chapter 6

# Conclusions

Classical approaches to time series analysis present severe limitations when analysing multidimensional sets of time series. A recent and very promising conceptual approach relies on mapping the time series to complex networks, where the large set of network science methodologies can help to gain new insights into the mapped time series. This thesis is focused precisely on the this research topic, trying to leverage the experience from both worlds. Our main goal is to contribute towards a rich complex networks based framework for the analysis of time series data.

This final chapter summarizes our main contributions and concludes with directions for future research.

## 6.1 Main Contributions

Our first main contribution was the proposal of using different network mapping concepts for creating a single richer set of features describing a network. Previous approaches only relied on one main concept, and here we show that different mappings complement each other, identifying different characteristics of time series. For instance, HVG captures more local characteristics, while quantiles capture the variability of the observations. By combining different mapping we therefore are able to better describe the original time series.

Our second major contribution, detailed in chapter 4, is a very detailed study of what each of five different topological metrics is capturing under each different mapping concept. This is the first time such a thorough and systematic study is made. For instance, we found that the average degree and clustering coefficient of NVGs tends to increase with increasing values of $\phi_1$ in AR and ARFIMA models, behavior which may be related to increased autocorrelation, while the average path length tends to decrease with higher autocorrelation. On the other hand, the average degree in HVGs seems to be able to distinguish discrete valued time series, such as INAR and HMM, while the number of communities separates the SETAR model from others.

A third major contribution, detailed in chapter 5, is a novel clustering methodology that takes advantage of all the structural metrics and mappings described before. We use our rich and vast feature set on a large set of model networks, employ dimensionality reduction techniques such as PCA and t-SNE, combined with standard clustering algorithms. Our results show the validity and discrimination power of our approach. For instance, we are able to distinguish synthetically generated time series of 17 different models (distinguishing linear from nonlinear, stationary from nonstationary, pseudo periodic from nonperiodic and autocorrelated from uncorrelated data), obtaining adjusted Rand indexes as high as 0.72 when using both visibility and quantile mappings, combined with t-SNE and k-means. We also show that we are competitive in clustering real time series when comparing with more traditional approaches, and for large time series we obtain significant gains in the quality of our clusters.

Our final contribution is precisely the fact that all our empirical tests used a very large and varied set of time series models, while previous work typically focuses on more smaller set of time series which are already much more well known. We selected 17 different models, traversing a very diverse range of linear an nonlinear models. We also experimented with shorter control time series and several real life data.

## 6.2 Future Work

The exploratory work done on this thesis provides a strong foundation for future work. While all our main goals were achieved, we already see many possible paths to continue our journey into contributing for this research area.

One angle we want to explore is the existing univariate mappings to multivariate and spatio-temporal contexts. For example, we intend to explore the notion of multilayer networks, which will allow to integrate in a single complex network system several different variables (for example, sea level, temperature and pressure could be represented on different layers in the same network) and spatial features (for example, nodes might be connected on a network layer that represents geographical proximity). Other possible approaches include the mapping to temporal networks that represents the state of the time series on different time intervals and the expansion of visibility concepts, for instance, considering directed edges or weighted connections that reflect the visibility angle (this might help in distinguishing increasing and decreasing trends, which are not captured by simple boolean visibility).

Another future plan is to expand the initial set of structural metrics that we used. In particular we want to find higher-order structures. We want to take advantage of our expertise in subgraph based metrics such as network motifs and graphlets [53], that really enrich the feature space (e.g. two networks might be indistinguishable by looking solely to their degree distribution, but be completely different when looking at their subgraphs distribution). We also aim to develop new metrics that can capture specific time series characteristics and to research different ways of combining all the metrics into lower dimensions (for example, we used PCA

and t-SNE on this thesis, and we will explore other options).

It also in our intentions to build an integrated framework incorporating all the developed methodologies. Given that we intend to tackle multivariate massive data, we need to produce efficient and scalable implementations of the associated algorithms (e.g. relevant network metrics might be estimated instead of exhaustively computed, allowing for finer trades between precision and time and space complexities). We intend to share with the community a ready to use tool that allows any practitioner to use our methods and take advantage of the available computational power (e.g. we consider the option of producing a parallel multicore version, which can give speedups even in normal commodity computers).

At the end of this thesis, we feel encouraged by the promising results obtained and firmly believe that time series using complex networks has the potential to really advance the field and help solving some of the open problems.

# Appendix A

# Characterization of Smaller Time Series using Topological Features

This appendix contains a set of tables similar to those presented in the section 4.1 with the particularity that the time series models corresponding to these measures of networks are relatively less length, that is, $T = 1000$. The objective is to see if the same topological characteristics of networks corresponding to series of length $T = 10000$ distinguish the same time series models as the topological characteristics corresponding to series of less length.

The topological features obtained from the 1700 time series are summarized in A.1 to A.5, each table reporting the mean and standard deviation (in brackets) of the standardized (across models) metrics. For easiness of interpretation, the columns of the tables are colored with a gradient based on the mean values: cells with a maximum value of 1 are colored red, cells with the minimum value 0 are colored white and the remainder with a hue of red color proportional to its value.

We verified that most of the measures allow to make the same characterization that we did for the time series models of 10000 length, with the exception of the number of communities and clustering coefficient of the corresponding 100-QGs. These differences lie in the fact that the `AR(1)-0.5` networks are now distinct of the `INAR` networks with a much higher number of communities. And now the highest values of clustering coefficient respect to networks `INAR` and `HMM` followed up the networks `ARIMA` and `ARFIMA0.9`.

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **WN** | 0.013 (0.002) | 0.339 (0.035) | 0.344 (0.101) | 0.478 (0.024) | 0.847 (0.049) |
| **AR(1)-0.5** | 0.005 (0.002) | 0.337 (0.045) | 0.322 (0.111) | 0.408 (0.026) | 0.838 (0.059) |
| **AR(1)0.5** | 0.037 (0.004) | 0.323 (0.037) | 0.360 (0.121) | 0.587 (0.031) | 0.865 (0.057) |
| **AR(1)0.9** | 0.104 (0.009) | 0.291 (0.042) | 0.271 (0.103) | 0.642 (0.037) | 0.873 (0.045) |
| **AR(2)** | 0.129 (0.006) | 0.294 (0.040) 1 | 0.557 (0.173) | 0.874 (0.047) | 0.858 (0.050) |
| **ARIMA** | 0.663 (0.132) | 0.176 (0.104) | 0.322 (0.212) | 0.304 (0.150) | 0.496 (0.221) |
| **ARFIMA-0.5** | 0.013 (0.003) | 0.335 (0.040) | 0.342 (0.107) | 0.437 (0.024) | 0.827 (0.058) |
| **ARFIMA0.5** | 0.083 (0.008) | 0.302 (0.045) | 0.329 (0.110) | 0.636 (0.034) | 0.865 (0.048) |
| **ARFIMA0.9** | 0.290 (0.031) | 0.225 (0.061) | 0.272 (0.141) | 0.532 (0.073) | 0.760 (0.085) |
| **ARCH** | 0.026 (0.005) | 0.267 (0.044) | 0.332 (0.125) | 0.421 (0.044) | 0.816 (0.056) |
| **GARCH1** | 0.015 (0.003) | 0.333 (0.045) | 0.358 (0.107) | 0.469 (0.028) | 0.830 (0.054) |
| **GARCH2** | 0.020 (0.004) | 0.294 (0.050) | 0.394 (0.127) | 0.428 (0.042) | 0.781 (0.066) |
| **EGARCH** | 0.020 (0.003) | 0.312 (0.042) | 0.344 (0.100) | 0.491 (0.028) | 0.852 (0.045) |
| **GJR-GARCH** | 0.136 (0.013) | 0.180 (0.048) | 0.283 (0.089) | 0.655 (0.070) | 0.895 (0.043) |
| **SETAR** | 0.010 (0.002) | 0.325 (0.040) | 0.289 (0.075) | 0.433 (0.022) | 0.860 (0.039) |
| **INAR** | 0.032 (0.005) | 0.321 (0.078) | 0.361 (0.113) | 0.436 (0.046) | 0.874 (0.058) |
| **HMM** | 0.021 (0.003) | 0.337 (0.047) | 0.319 (0.089) | 0.483 (0.028) | 0.879 (0.048) |

Table A.1: Table of mean values of the 100 instances of each time series model for each topological metric, resulting from NVGs. The standard deviations are presented in parentheses.

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **WN** | 0.976 (0.006) | 0.012 (0.005) | 0.047 (0.014) | 0.632 (0.012) | 0.319 (0.111) |
| **AR(1)-0.5** | 0.977 (0.005) | 0.012 (0.005) | 0.028 (0.012) | 0.589 (0.011) | 0.359 (0.104) |
| **AR(1)0.5** | 0.975 (0.006) | 0.023 (0.006) | 0.061 (0.021) | 0.714 (0.015) | 0.352 (0.132) |
| **AR(1)0.9** | 0.962 (0.010) | 0.073 (0.012) | 0.047 (0.021) | 0.797 (0.012) | 0.636 (0.105) |
| **AR(2)** | 0.965 (0.009) | 0.056 (0.008) | 0.128 (0.040) | 0.951 (0.011) | 0.337 (0.109) |
| **ARIMA** | 0.792 (0.081) | 0.499 (0.151) | 0.049 (0.013) | 0.968 (0.013) | 0.888 (0.044) |
| **ARFIMA-0.5** | 0.974 (0.007) | 0.020 (0.007) | 0.034 (0.017) | 0.611 (0.010) | 0.389 (0.122) |
| **ARFIMA0.5** | 0.963 (0.010) | 0.062 (0.014) | 0.056 (0.026) | 0.797 (0.013) | 0.569 (0.140) |
| **ARFIMA0.9** | 0.920 (0.026) | 0.204 (0.047) | 0.041 (0.015) | 0.899 (0.014) | 0.827 (0.062) |
| **ARCH** | 0.975 (0.006) | 0.020 (0.007) | 0.040 (0.015) | 0.642 (0.016) | 0.400 (0.129) |
| **GARCH1** | 0.975 (0.005) | 0.017 (0.006) | 0.044 (0.017) | 0.635 (0.015) | 0.350 (0.104) |
| **GARCH2** | 0.998 (0.001) | 0.073 (0.010) | 0.577 (0.132) | 0.624 (0.004) | 0.599 (0.097) |
| **EGARCH** | 0.977 (0.006) | 0.012 (0.005) | 0.041 (0.015) | 0.620 (0.010) | 0.308 (0.126) |
| **GJR-GARCH** | 0.969 (0.008) | 0.038 (0.008) | 0.046 (0.018) | 0.663 (0.021) | 0.592 (0.105) |
| **SETAR** | 0.977 (0.006) | 0.007 (0.004) | 0.021 (0.012) | 0.542 (0.010) | 0.549 (0.107) |
| **INAR** | 0.066 (0.022) | 0.072 (0.015) | 0.069 (0.026) | 0.085 (0.037) | 0.633 (0.101) |
| **HMM** | 0.647 (0.019) | 0.030 (0.008) | 0.047 (0.018) | 0.495 (0.019) | 0.555 (0.121) |

Table A.2: Table of mean values of the 100 instances of each time series model for each topological metric, resulting from HVGs. The standard deviations are presented in parentheses.

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **WN** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.143 (0.024) |
| **AR(1)-0.5** | 0.000 (0.000) | 0.003 (0.004) | 0.000 (0.000) | 1.000 (0.003) | 0.051 (0.021) |
| **AR(1)0.5** | 0.000 (0.000) | 0.007 (0.005) | 0.243 (0.149) | 0.995 (0.010) | 0.374 (0.045) |
| **AR(1)0.9** | 0.000 (0.000) | 0.155 (0.027) | 0.343 (0.088) | 0.777 (0.031) | 0.821 (0.080) |
| **AR(2)** | 0.000 (0.000) | 0.107 (0.013) | 0.333 (0.000) | 0.827 (0.021) | 0.776 (0.053) |
| **ARIMA** | 0.000 (0.000) | 0.983 (0.047) | 0.063 (0.131) | 0.014 (0.060) | 0.192 (0.098) |
| **ARFIMA-0.5** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.144 (0.032) |
| **ARFIMA0.5** | 0.000 (0.000) | 0.120 (0.025) | 0.337 (0.089) | 0.810 (0.034) | 0.811 (0.085) |
| **ARFIMA0.9** | 0.000 (0.000) | 0.691 (0.125) | 0.257 (0.149) | 0.455 (0.090) | 0.644 (0.077) |
| **ARCH** | 0.000 (0.000) | 0.000 (0.001) | 0.030 (0.096) | 1.000 (0.000) | 0.222 (0.029) |
| **GARCH1** | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 1.000 (0.000) | 0.168 (0.030) |
| **GARCH2** | 0.000 (0.000) | 0.000 (0.000) | 0.003 (0.033) | 1.000 (0.000) | 0.192 (0.032) |
| **EGARCH** | 0.000 (0.000) | 0.000 (0.001) | 0.000 (0.000) | 1.000 (0.000) | 0.143 (0.028) |
| **GJR-GARCH** | 0.000 (0.000) | 0.226 (0.018) | 0.437 (0.175) | 0.973 (0.023) | 0.750 (0.101) |
| **SETAR** | 0.000 (0.000) | 0.044 (0.010) | 0.330 (0.033) | 0.997 (0.009) | 0.275 (0.037) |
| **INAR** | 0.673 (0.047) | 0.020 (0.011) | 0.000 (0.000) | 0.964 (0.039) | 0.609 (0.022) |
| **HMM** | 0.003 (0.019) | 0.001 (0.002) | 0.043 (0.113) | 1.000 (0.000) | 0.255 (0.031) |

Table A.3: Table of mean values of the 100 instances of each time series model for each topological metric, resulting from 10-QGs. The standard deviations are presented in parentheses.

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **WN** | 0.000 (0.000) | 0.043 (0.000) | 0.110 (0.046) | 0.401 (0.013) | 0.174 (0.028) |
| **AR(1)-0.5** | 0.000 (0.000) | 0.044 (0.000) | 0.025 (0.143) | 0.341 (0.016) | 0.013 (0.006) |
| **AR(1)0.5** | 0.000 (0.000) | 0.044 (0.000) | 0.060 (0.023) | 0.379 (0.015) | 0.303 (0.041) |
| **AR(1)0.9** | 0.000 (0.000) | 0.072 (0.004) | 0.115 (0.037) | 0.454 (0.020) | 0.691 (0.043) |
| **AR(2)** | 0.000 (0.000) | 0.060 (0.001) | 0.089 (0.029) | 0.429 (0.018) | 0.599 (0.028) |
| **ARIMA** | 0.000 (0.000) | 0.707 (0.139) | 0.203 (0.050) | 0.347 (0.108) | 0.732 (0.097) |
| **ARFIMA-0.5** | 0.000 (0.000) | 0.043 (0.000) | 0.106 (0.047) | 0.395 (0.013) | 0.179 (0.025) |
| **ARFIMA0.5** | 0.000 (0.000) | 0.064 (0.005) | 0.097 (0.031) | 0.439 (0.020) | 0.645 (0.048) |
| **ARFIMA0.9** | 0.000 (0.000) | 0.235 (0.040) | 0.200 (0.049) | 0.530 (0.018) | 0.934 (0.029) |
| **ARCH** | 0.000 (0.000) | 0.043 (0.000) | 0.103 (0.063) | 0.389 (0.013) | 0.169 (0.035) |
| **GARCH1** | 0.000 (0.000) | 0.043 (0.000) | 0.107 (0.051) | 0.396 (0.012) | 0.186 (0.029) |
| **GARCH2** | 0.000 (0.000) | 0.043 (0.000) | 0.106 (0.055) | 0.393 (0.012) | 0.178 (0.030) |
| **EGARCH** | 0.000 (0.000) | 0.043 (0.000) | 0.103 (0.054) | 0.398 (0.013) | 0.172 (0.028) |
| **GJR-GARCH** | 0.000 (0.000) | 0.092 (0.004) | 0.119 (0.029) | 0.282 (0.018) | 0.631 (0.036) |
| **SETAR** | 0.000 (0.000) | 0.050 (0.001) | 0.065 (0.027) | 0.359 (0.015) | 0.199 (0.037) |
| **INAR** | 0.964 (0.063) | 0.006 (0.004) | 0.000 (0.000) | 0.875 (0.051) | 0.392 (0.011) |
| **HMM** | 0.275 (0.015) | 0.010 (0.002) | 0.043 (0.010) | 0.882 (0.025) | 0.189 (0.032) |

Table A.4: Table of mean values of the 100 instances of each time series model for each topological metric, resulting from 50-QGs. The standard deviations are presented in parentheses.

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **WN** | 0.000 (0.000) | 0.054 (0.000) | 0.117 (0.041) | 0.035 (0.005) | 0.225 (0.028) |
| **AR(1)-0.5** | 0.000 (0.000) | 0.055 (0.000) | 0.466 (0.219) | 0.016 (0.006) | 0.061 (0.037) |
| **AR(1)0.5** | 0.000 (0.000) | 0.055 (0.000) | 0.055 (0.039) | 0.045 (0.007) | 0.252 (0.029) |
| **AR(1)0.9** | 0.000 (0.000) | 0.074 (0.003) | 0.076 (0.022) | 0.154 (0.015) | 0.616 (0.039) |
| **AR(2)** | 0.000 (0.000) | 0.065 (0.001) | 0.062 (0.020) | 0.116 (0.008) | 0.515 (0.026) |
| **ARIMA** | 0.000 (0.000) | 0.610 (0.158) | 0.197 (0.037) | 0.488 (0.028) | 0.886 (0.061) |
| **ARFIMA-0.5** | 0.000 (0.000) | 0.054 (0.000) | 0.110 (0.035) | 0.034 (0.005) | 0.226 (0.018) |
| **ARFIMA0.5** | 0.000 (0.000) | 0.068 (0.003) | 0.066 (0.022) | 0.130 (0.018) | 0.561 (0.047) |
| **ARFIMA0.9** | 0.000 (0.000) | 0.190 (0.029) | 0.149 (0.033) | 0.384 (0.037) | 0.918 (0.038) |
| **ARCH** | 0.000 (0.000) | 0.055 (0.000) | 0.134 (0.049) | 0.036 (0.006) | 0.210 (0.024) |
| **GARCH1** | 0.000 (0.000) | 0.054 (0.000) | 0.120 (0.036) | 0.034 (0.005) | 0.225 (0.017) |
| **GARCH2** | 0.000 (0.000) | 0.054 (0.000) | 0.113 (0.038) | 0.036 (0.006) | 0.222 (0.019) |
| **EGARCH** | 0.000 (0.000) | 0.054 (0.000) | 0.113 (0.033) | 0.034 (0.005) | 0.221 (0.019) |
| **GJR-GARCH** | 0.000 (0.000) | 0.082 (0.003) | 0.077 (0.019) | 0.088 (0.010) | 0.554 (0.033) |
| **SETAR** | 0.000 (0.000) | 0.058 (0.001) | 0.129 (0.056) | 0.028 (0.007) | 0.200 (0.026) |
| **INAR** | 0.872 (0.040) | 0.007 (0.003) | 0.000 (0.002) | 0.832 (0.046) | 0.292 (0.010) |
| **HMM** | 0.284 (0.010) | 0.011 (0.001) | 0.027 (0.008) | 0.818 (0.025) | 0.106 (0.031) |

Table A.5: Table of mean values of the 100 instances of each time series model for each topological metric, resulting from 100-QGs. The standard deviations are presented in parentheses.

# Appendix B

# Variations of Topological Features for Series of Different Lengths

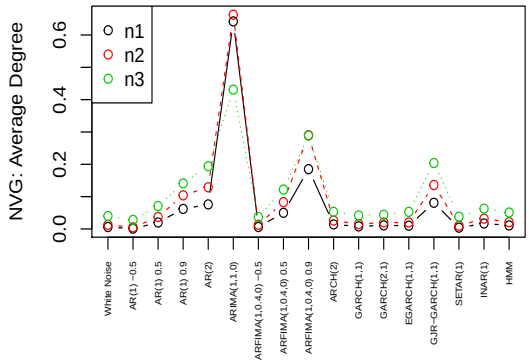
This appendix contains a very brief overview of how the global topological metrics of networks vary with the variation of different lengths of time series that networks represent. Here we analyze three different lengths, namely $T = 10000$, $T = 1000$ and $T = 100$, which in the plots correspond to the lines $n1$, $n2$ and $n3$, respectively. The plots represent the mean values of the respective metric calculated for the 100 instances of networks of each type of corresponding time series model.

Through these plots we can easily verify what metrics of which type of mapping varies and for what type of network. For example, the metrics corresponding to the 100-QGs (figure B.5) produce quite different values for the set $n3$, because actually the number of qunatiles chosen in this mapping is equal to the number of observations of the time series which in this type of mapping does not make sense, as explained in section 3.3. We also found large differences between the sets $n1$ and $n2$ for the number of communities and clustering coefficient metrics.

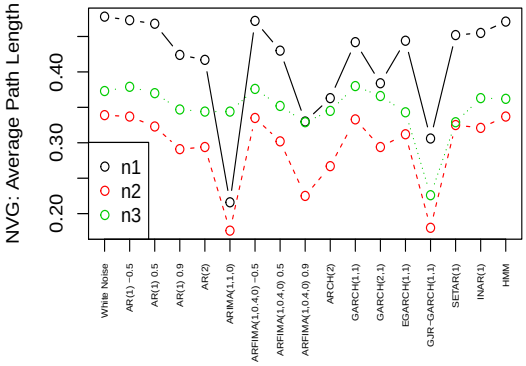
The values corresponding to the NVGs in some cases are different, however, we are not completely interested in that the values are the same, but that they preserve the difference of values for the different types of networks, that is, that the differences between the values of the same metrics obtain to distinguish the same sets of network types for the different sets $n1$, $n2$ and $n3$. In the case of NVGs this happens.

Already HVGs show that the metric number of communities and modularity are not good for the classification of time series, since it vary with different lengths. On the other hand the 10-QGs seem to be pretty good on any of the metrics, because 10 is a relatively small number for the three sets, as opposed to 50 and 100.

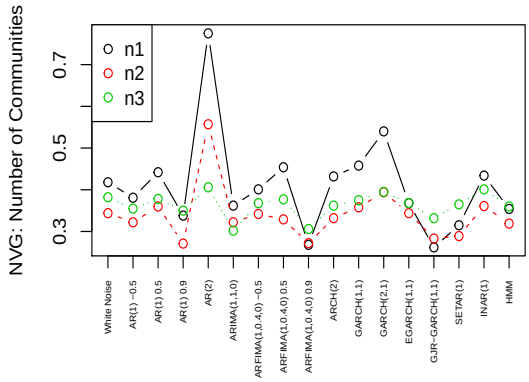
It should be mentioned that we are aware that these results are a brief analysis, that is, a very detailed survey was not made to consider these conclusive results.



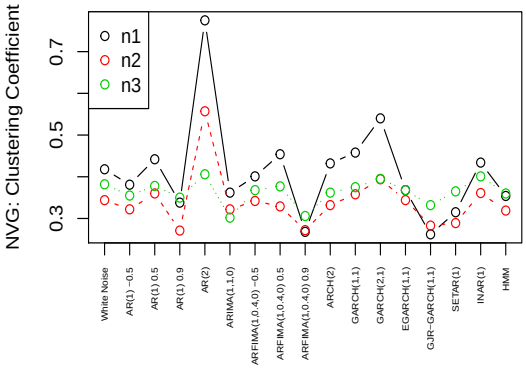

(a) Average Degree



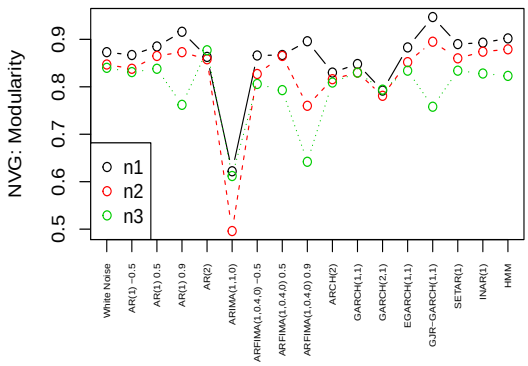

(b) Average Path Length




(c) Number of Communities

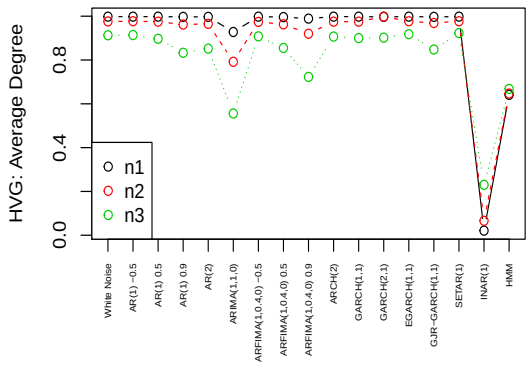



(d) Clustering Coefficient

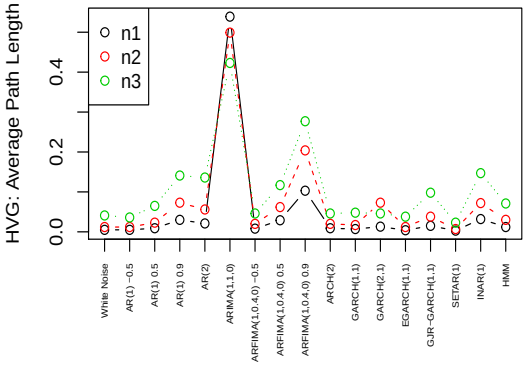



(e) Modularity

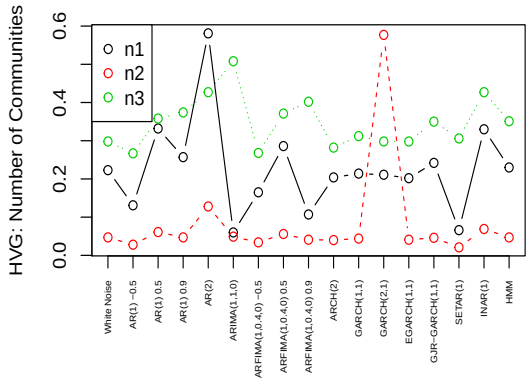
Figure B.1: Variations of the topological measures of NVGs corresponding to time series of different lengths.



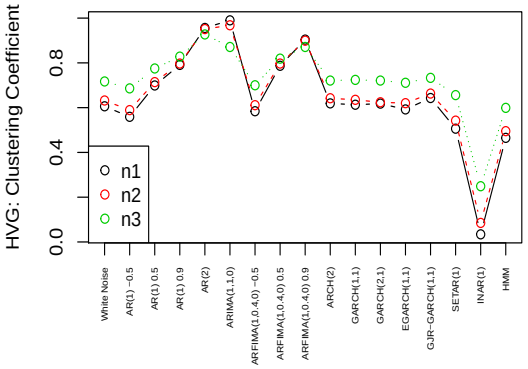

(a) Average Degree



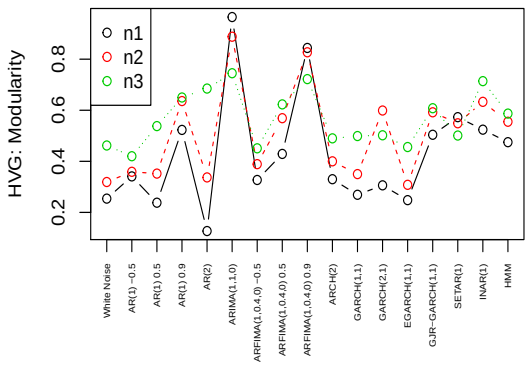

(b) Average Path Length




(c) Number of Communities

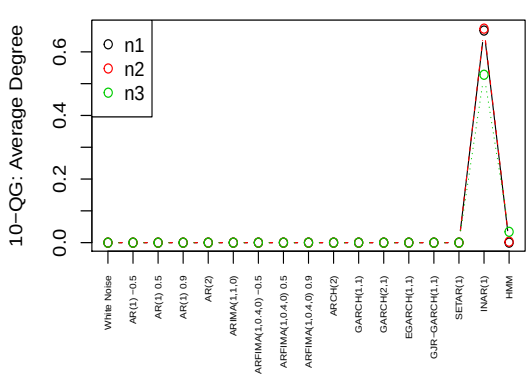



(d) Clustering Coefficient

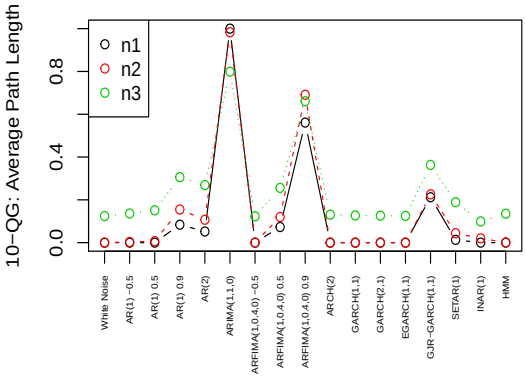



(e) Modularity

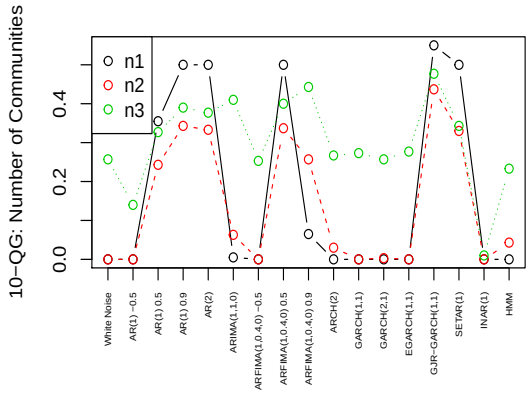
Figure B.2: Variations of the topological measures of HVGs corresponding to time series of different lengths.



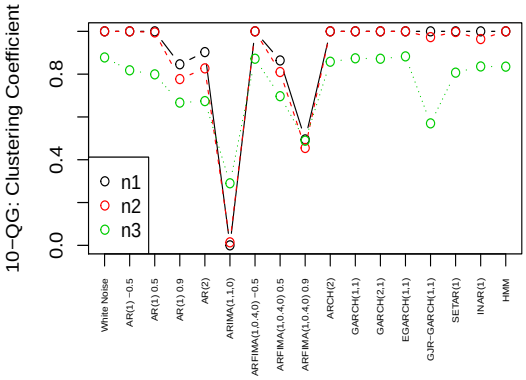

(a) Average Degree



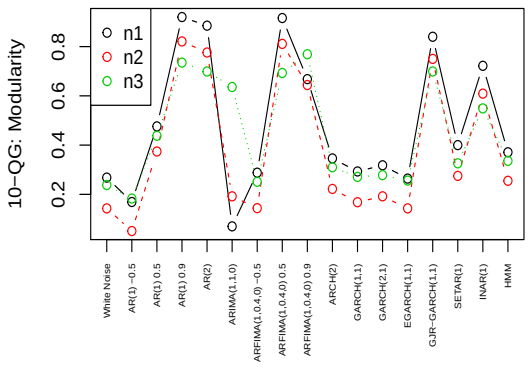

(b) Average Path Length




(c) Number of Communities

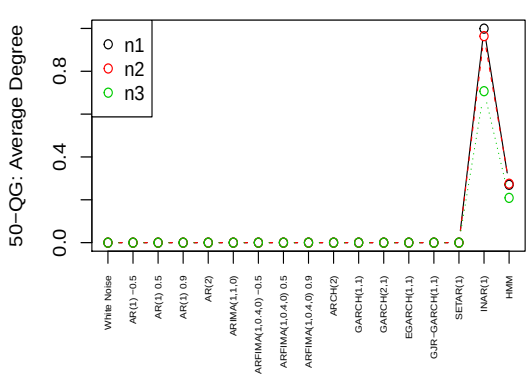



(d) Clustering Coefficient

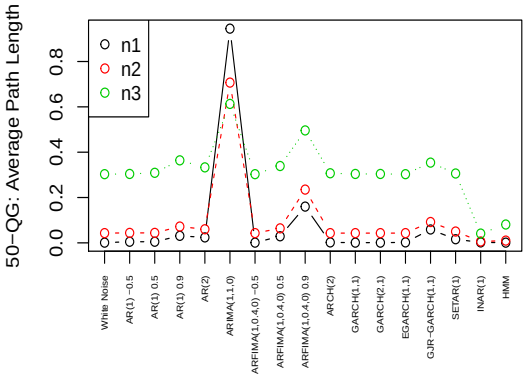



(e) Modularity

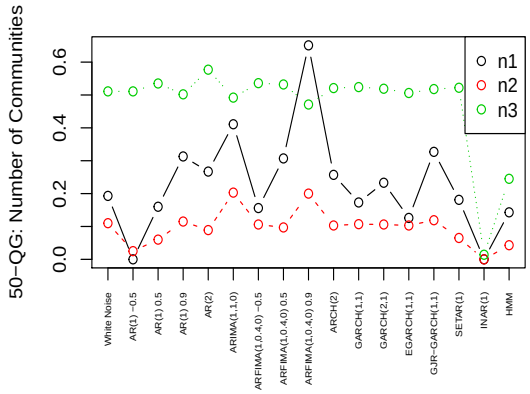
Figure B.3: Variations of the topological measures of 10-QGs corresponding to time series of different lengths.



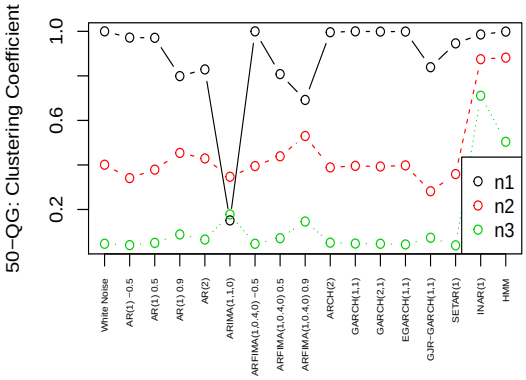

(a) Average Degree



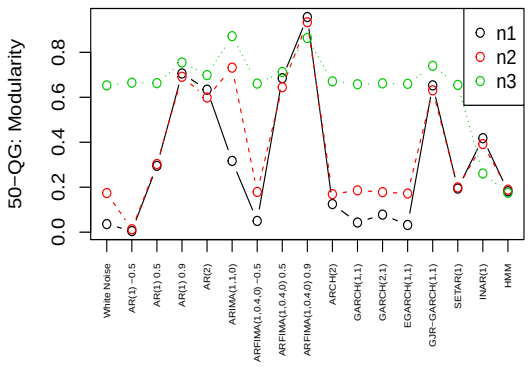

(b) Average Path Length




(c) Number of Communities

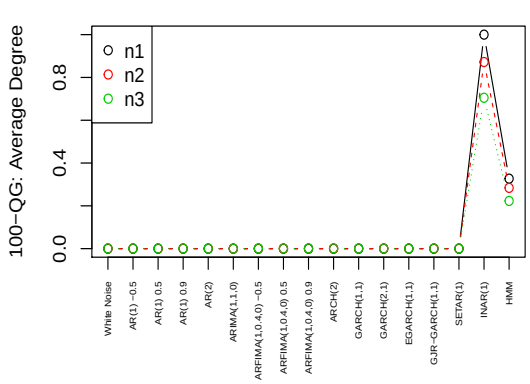



(d) Clustering Coefficient

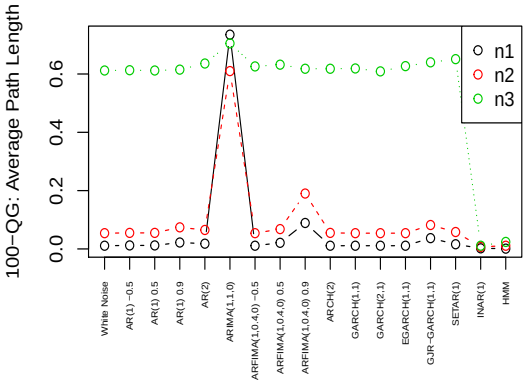



(e) Modularity

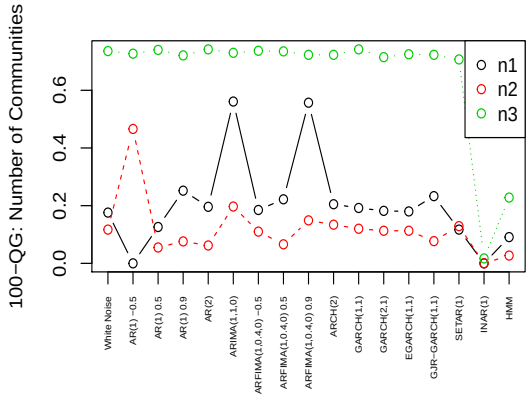
Figure B.4: Variations of the topological measures of 50-QGs corresponding to time series of different lengths.



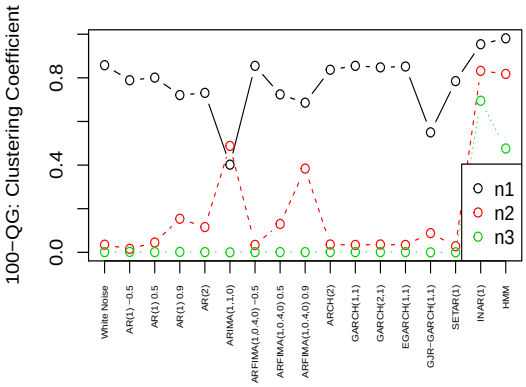

(a) Average Degree



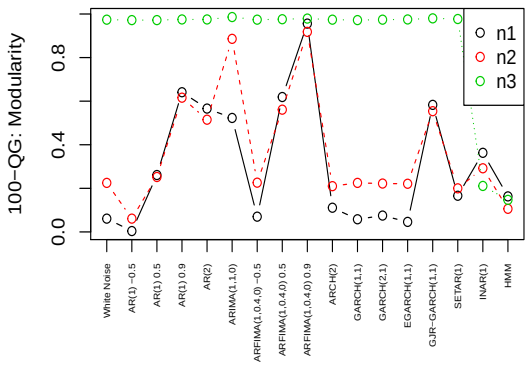

(b) Average Path Length




(c) Number of Communities




(d) Clustering Coefficient




(e) Modularity

Figure B.5: Variations of the topological measures of 100-QGs corresponding to time series of different lengths.

# Appendix C

# Topological Metrics from Trend Time Series

This appendix contains a set of tables with the mean and standard deviation (in brackets) of the standardized (across time series of different trends) metrics.

The objective is to see if different types of trends are distinguishable using the five global topological metrics of networks of different mappings (NVG, HVG, 10-QG, 50-QG and 100-QG). For this we use 100 instances of three types of time series of length $T = 10000$ with different trend, namely, `ARFIMA0.9` that has local trending behavior, `ARIMA` the nonstationary series with stochastic trends, and `D_Trend` which is a series with deterministic trend. Most of the calculated metrics can distinguish the three types of trend.

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **ARFIMA(1,0.4,0) 0.9** | 0.185 (0.007) | 0.330 (0.054) | 0.268 (0.102) | 0.656 (0.026) | 0.896 (0.040) |
| **ARIMA(1,1,0)** | 0.642 (0.109) | 0.216 (0.124) | 0.362 (0.217) | 0.205 (0.089) | 0.622 (0.174) |
| **D_Trend** | 0.442 (0.205) | 0.431 (0.207) | 0.526 (0.202) | 0.481 (0.211) | 0.551 (0.229) |

Table C.1: Table of mean values of the 100 instances of each trend time series for each topological metric, resulting from NVGs. The standard deviations are presented in parentheses.

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **ARFIMA(1,0.4,0) 0.9** | 0.989 (0.003) | 0.103 (0.012) | 0.107 (0.045) | 0.904 (0.004) | 0.844 (0.038) |
| **ARIMA(1,1,0)** | 0.928 (0.031) | 0.539 (0.151) | 0.060 (0.026) | 0.990 (0.004) | 0.965 (0.017) |
| **D_Trend** | 0.552 (0.222) | 0.446 (0.203) | 0.530 (0.229) | 0.497 (0.201) | 0.444 (0.250) |

Table C.2: Table of mean values of the 100 instances of each trend time series for each topological metric, resulting from HVGs. The standard deviations are presented in parentheses.

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **ARFIMA(1,0.4,0) 0.9** | 0.000 (0.000) | 0.561 (0.076) | 0.065 (0.169) | 0.495 (0.033) | 0.668 (0.038) |
| **ARIMA(1,1,0)** | 0.000 (0.000) | 1.000 (0.000) | 0.005 (0.050) | 0.000 (0.000) | 0.070 (0.032) |
| **D_Trend** | 0.200 (0.000) | 0.660 (0.404) | 1.000 (0.000) | 0.000 (0.000) | 0.453 (0.194) |

Table C.3: Table of mean values of the 100 instances of each trend time series for each topological metric, resulting from 10-QGs. The standard deviations are presented in parentheses.

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **ARFIMA(1,0.4,0) 0.9** | 0.000 (0.000) | 0.160 (0.017) | 0.651 (0.155) | 0.692 (0.008) | 0.957 (0.016) |
| **ARIMA(1,1,0)** | 0.000 (0.000) | 0.945 (0.059) | 0.411 (0.150) | 0.151 (0.132) | 0.317 (0.073) |
| **D_Trend** | 0.040 (0.000) | 0.208 (0.207) | 0.505 (0.271) | 0.000 (0.000) | 0.592 (0.198) |

Table C.4: Table of mean values of the 100 instances of each trend time series for each topological metric, resulting from 50-QGs. The standard deviations are presented in parentheses.

| Models | Average Degree | Average Path Length | Number of Communities | Clustering Coefficient | Modularity |
|---|---|---|---|---|---|
| **ARFIMA(1,0.4,0) 0.9** | 0.000 (0.000) | 0.089 (0.009) | 0.557 (0.115) | 0.686 (0.004) | 0.956 (0.015) |
| **ARIMA(1,1,0)** | 0.000 (0.000) | 0.735 (0.124) | 0.561 (0.120) | 0.402 (0.119) | 0.523 (0.096) |
| **D_Trend** | 0.020 (0.000) | 0.414 (0.208) | 0.530 (0.182) | 0.000 (0.000) | 0.523 (0.193) |

Table C.5: Table of mean values of the 100 instances of each trend time series for each topological metric, resulting from 100-QGs. The standard deviations are presented in parentheses.

# Appendix D

# Clustering of Time Series Models: better $k$

This appendix present in the tables D.1 and D.2 the results obtained in the clustering analysis similar to those presented in section 5.2 for the first and second experience, respectively, with the particularity that here we do not provide a priori the true value of $k$ to the $k$-means algorithm. This $k$ is calculated iteratively and chosen the one that obtained better results from the adjusted Rand index.

The results refer to the evaluation metrics and the better $k$ calculated for the data set whose corresponding time series have length 10000 (columns 2, 3 and 4), and for the data set whose corresponding time series have length 1000 (columns 5, 6 and 7). The colors represent the 3 maximum values of the corresponding column, with the darker color highlighting the maximum value and the lighter color the third maximum value.

| Mappings | $n_1 = 10000$ | | | $n_2 = 1000$ | | |
|---|---|---|---|---|---|---|
| | **Adjusted Rand Index** | **Average Silhouette** | $k$ | **Adjusted Rand Index** | **Average Silhouette** | $k$ |
| **NVG** | 0.19 | 0.34 | 16 | 0.15 | 0.35 | 9 |
| **HVG** | 0.40 | 0.49 | 17 | 0.24 | 0.42 | 15 |
| **Q10** | **0.59** | **0.66** | **17** | **0.39** | **0.51** | **13** |
| **Q50** | **0.43** | **0.60** | **17** | **0.45** | **0.48** | **16** |
| **Q100** | **0.56** | **0.62** | **17** | 0.36 | 0.46 | 13 |
| **NVG-HVG** | 0.46 | 0.48 | 15 | 0.29 | 0.39 | 13 |
| **NVG-Q10** | 0.51 | 0.56 | 17 | 0.31 | 0.39 | 11 |
| **NVG-Q50** | 0.49 | 0.48 | 17 | **0.46** | **0.45** | **15** |
| **NVG-Q100** | 0.55 | 0.55 | 15 | 0.36 | 0.41 | 17 |
| **HVG-Q10** | 0.48 | 0.53 | 17 | 0.40 | 0.45 | 15 |
| **HVG-Q50** | 0.51 | 0.51 | 15 | 0.44 | 0.44 | 15 |
| **HVG-Q100** | 0.55 | 0.52 | 17 | 0.36 | 0.42 | 15 |
| **Q10-Q50** | 0.53 | 0.59 | 15 | **0.46** | **0.51** | **17** |
| **Q10-Q100** | 0.53 | 0.58 | 17 | **0.47** | **0.49** | **15** |
| **Q50-Q100** | 0.53 | 0.58 | 17 | **0.37** | **0.48** | **15** |
| **NVG-HVG-Q10** | 0.40 | 0.57 | 17 | 0.41 | 0.38 | 17 |
| **NVG-HVG-Q50** | 0.59 | 0.48 | 17 | 0.40 | 0.40 | 15 |
| **NVG-HVG-Q100** | **0.61** | **0.51** | **17** | 0.37 | 0.40 | 13 |
| **NVG-Q10-Q50** | 0.54 | 0.53 | 15 | **0.46** | **0.46** | **17** |
| **NVG-Q10-Q100** | 0.52 | 0.52 | 17 | **0.45** | **0.44** | **15** |
| **NVG-Q50-Q100** | 0.55 | 0.53 | 17 | 0.44 | 0.44 | 17 |
| **HVG-Q10-Q50** | 0.50 | 0.52 | 17 | 0.37 | 0.41 | 17 |
| **HVG-Q10-Q100** | 0.53 | 0.50 | 17 | 0.39 | 0.44 | 17 |
| **HVG-Q50-Q100** | **0.63** | **0.57** | **15** | **0.45** | **0.48** | **15** |
| **Q10-Q50-Q100** | 0.51 | 0.55 | 15 | 0.44 | 0.47 | 17 |
| **NVG-HVG-Q10-Q50** | 0.53 | 0.57 | 15 | 0.43 | 0.44 | 15 |
| **NVG-HVG-Q10-Q100** | 0.53 | 0.50 | 17 | 0.40 | 0.42 | 15 |
| **NVG-HVG-Q50-Q100** | **0.62** | **0.56** | **15** | 0.43 | 0.42 | 15 |
| **NVG-Q10-Q50-Q100** | 0.51 | 0.54 | 15 | **0.45** | **0.43** | **15** |
| **HVG-Q10-Q50-Q100** | 0.52 | 0.55 | 17 | **0.45** | **0.460** | **15** |
| **NVG-HVG-Q10-Q50-Q100** | 0.59 | 0.54 | **15** | 0.42 | 0.43 | 17 |

Table D.1: Clustering evaluation metrics for the different clustering analysis performed using PCA as a dimensionality reduction technique. The results refer to the evaluation metrics and the better $k$ calculated for the dataset with time series of length 10000, columns 2, 3 and 4 and length 1000, columns 5, 6 and 7.

| Mappings | $n_1 = 10000$ | | | $n_2 = 1000$ | | |
|---|---|---|---|---|---|---|
| | **Adjusted Rand Index** | **Average Silhouette** | $k$ | **Adjusted Rand Index** | **Average Silhouette** | $k$ |
| **NVG** | 0.25 | 0.49 | 17 | 0.21 | 0.45 | 17 |
| **HVG** | 0.57 | 0.60 | 17 | 0.35 | 0.54 | 17 |
| **Q10** | 0.62 | 0.64 | 16 | **0.51** | **0.62** | **17** |
| **Q50** | **0.55** | **0.68** | **17** | 0.54 | 0.60 | 15 |
| **Q100** | **0.56** | **0.70** | **17** | **0.44** | **0.63** | **16** |
| **NVG-HVG** | 0.54 | 0.58 | 17 | 0.39 | 0.52 | 16 |
| **NVG-Q10** | 0.62 | 0.63 | 17 | 0.47 | 0.51 | 16 |
| **NVG-Q50** | 0.59 | 0.64 | 17 | **0.57** | **0.58** | **17** |
| **NVG-Q100** | **0.66** | **0.70** | **17** | 0.50 | 0.53 | 17 |
| **HVG-Q10** | 0.61 | 0.65 | 17 | 0.52 | 0.59 | 17 |
| **HVG-Q50** | 0.60 | 0.65 | 14 | **0.60** | **0.59** | **17** |
| **HVG-Q100** | 0.63 | 0.66 | 17 | **0.57** | **0.63** | **16** |
| **Q10-Q50** | 0.64 | 0.67 | 17 | 0.56 | 0.60 | 17 |
| **Q10-Q100** | 0.58 | 0.67 | 14 | **0.57** | **0.61** | **17** |
| **Q50-Q100** | 0.65 | 0.67 | 17 | 0.48 | 0.60 | 16 |
| **NVG-HVG-Q10** | 0.61 | 0.63 | 17 | 0.54 | 0.54 | 16 |
| **NVG-HVG-Q50** | 0.67 | 0.63 | 17 | 0.50 | 0.56 | 17 |
| **NVG-HVG-Q100** | **0.69** | **0.65** | **17** | 0.51 | 0.53 | 16 |
| **NVG-Q10-Q50** | **0.63** | 0.63 | 17 | **0.60** | **0.58** | **16** |
| **NVG-Q10-Q100** | 0.61 | 0.63 | 17 | 0.54 | 0.58 | 17 |
| **NVG-Q50-Q100** | **0.71** | **0.66** | **17** | 0.56 | 0.60 | 17 |
| **HVG-Q10-Q50** | 0.63 | 0.61 | 17 | 0.56 | 0.60 | 17 |
| **HVG-Q10-Q100** | 0.59 | 0.61 | 17 | 0.53 | 0.56 | 17 |
| **HVG-Q50-Q100** | **0.67** | **0.67** | **16** | **0.59** | **0.61** | **17** |
| **Q10-Q50-Q100** | **0.63** | **0.70** | **17** | 0.56 | 0.60 | 17 |
| **NVG-HVG-Q10-Q50** | 0.65 | 0.62 | 17 | 0.54 | 0.57 | 17 |
| **NVG-HVG-Q10-Q100** | 0.67 | 0.61 | 17 | 0.52 | 0.54 | 17 |
| **NVG-HVG-Q50-Q100** | **0.72** | **0.68** | **17** | 0.53 | 0.54 | 17 |
| **NVG-Q10-Q50-Q100** | **0.67** | **0.68** | **16** | **0.60** | **0.58** | **17** |
| **HVG-Q10-Q50-Q100** | 0.64 | 0.62 | 17 | **0.59** | **0.59** | **17** |
| **NVG-HVG-Q10-Q50-Q100** | 0.66 | 0.62 | 17 | 0.55 | 0.57 | 17 |

Table D.2: Clustering evaluation metrics for the different clustering analysis performed using t-SNE as a dimensionality reduction technique. The results refer to the evaluation metrics and the better $k$ calculated for the dataset with time series of length 10000, columns 2, 3 and 4 and length 1000, columns 5, 6 and 7.

# Bibliography


[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Luis Antonio Aguirre. *Introdução à identificação de sistemas–Técnicas lineares e não-lineares aplicadas a sistemas reais*. Editora UFMG, 2004.

[3] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[4] Robert J Alcock, Yannis Manolopoulos, et al. Time-series similarity queries employing a feature-based approach. In *7th Hellenic conference on informatics*, pages 27–29, 1999.

[5] Rodolfo Baggio and Ruggero Sainaghi. Mapping time series into networks as a tool to assess the complex dynamics of tourism systems. *Tourism Management*, 54:23–33, 2016.

[6] Albert-László Barabási. *Network Science*. Cambridge University Press, 2016.

[7] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. The architecture of complex weighted networks: Measurements and models. In *Large Scale Structure And Dynamics Of Complex Networks: From Information Technology to Finance and Natural Science*, pages 67–92. World Scientific, 2007.

[8] IV Bezsudnov and AA Snarskii. From the time series to the complex networks: The parametric natural visibility graph. *Physica A: Statistical Mechanics and its Applications*, 414:53–60, 2014.

[9] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[10] Andriana Campanharo and Fernando Ramos. Distinguishing different dynamics in electroencephalographic time series through a complex network approach. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 5(1), 2017.

[11] Andriana SLO Campanharo, M Irmak Sirer, R Dean Malmgren, Fernando M Ramos, and Luís A Nunes Amaral. Duality between time series and networks. *PloS one*, 6(8):e23378, 2011.

[12] Ricardo JGB Campello. A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters*, 28(7):833–841, 2007.

[13] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.

[14] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[15] L da F Costa, Francisco A Rodrigues, Gonzalo Travieso, and Paulino Ribeiro Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in physics*, 56 (1):167–242, 2007.

[16] Luciano da Fontoura Costa, Osvaldo N Oliveira Jr, Gonzalo Travieso, Francisco Aparecido Rodrigues, Paulino Ribeiro Villas Boas, Lucas Antiqueira, Matheus Palhares Viana, and Luis Enrique Correa Rocha. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. *Advances in Physics*, 60(3):329–412, 2011.

[17] Jonathan D Cryer and Kung-Sik Chan. *Time Series Analysis With Applications in R*. New York: Springer, 2008.

[18] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[19] Andriana Susana Lopes de Oliveira Campanharo and Fernando Manuel Ramos. Quantile graphs for the characterization of chaotic dynamics in time series. In *Complex Systems (WCCS), 2015 Third World Conference on*, pages 1–4. IEEE, 2015.

[20] Reik V Donner, Yong Zou, Jonathan F Donges, Norbert Marwan, and Jürgen Kurths. Recurrence networks—a novel paradigm for nonlinear time series analysis. *New Journal of Physics*, 12(3):033025, 2010.

[21] Philippe Esling and Carlos Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012.

[22] Jianqing Fan and Qiwei Yao. *Nonlinear time series: nonparametric and parametric methods*. Springer Science & Business Media, 2008.

[23] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[24] Philip Hans Franses and Dick Van Dijk. *Non-linear time series models in empirical finance*. Cambridge University Press, 2000.

[25] Zhong-Ke Gao, Michael Small, and Jürgen Kurths. Complex network analysis of time series. *EPL (Europhysics Letters)*, 116(5):50001, 2017.

[26] Zhongke Gao and Ningde Jin. Complex network from time series based on phase space reconstruction. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(3):033137, 2009.

[27] Alexios Ghalanos. *rugarch: Univariate GARCH models.*, 2018. R package version 1.4-0.

[28] Christian Hennig, Marina Meila, Fionn Murtagh, and Roberto Rocci. *Handbook of cluster analysis*. CRC Press, 2015.

[29] Rob Hyndman. Measuring time series characteristics. Online, 2012. May 2018.

[30] Jacopo Iacovacci and Lucas Lacasa. Sequential motif profile of natural visibility graphs. *Physical Review E*, 94(5):052309, 2016.

[31] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8): 651–666, 2010.

[32] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

[33] Argyro Kampouraki, George Manis, and Christophoros Nikou. Heartbeat time series classification with support vector machines. *IEEE Transactions on Information Technology in Biomedicine*, 13(4):512–518, 2009.

[34] Yanfei Kang, Rob J Hyndman, and Kate Smith-Miles. Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting*, 33(2): 345–358, 2017.

[35] Lucas Lacasa and Raul Toral. Description of stochastic and chaotic series using visibility graphs. *Physical Review E*, 82(3):036120, 2010.

[36] Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nuno. From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences*, 105(13):4972–4975, 2008.

[37] Lucas Lacasa, Bartolo Luque, Jordi Luque, and Juan Carlos Nuno. The visibility graph: A new method for estimating the hurst exponent of fractional brownian motion. *EPL (Europhysics Letters)*, 86(3):30001, 2009.

[38] Xin Lan, Hongming Mo, Shiyu Chen, Qi Liu, and Yong Deng. Fast transformation from time series to visibility graphs. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(8):083105, 2015.

[39] Daoyuan Li, Jessica Lin, Tegawendé François D Assise Bissyande, Jacques Klein, and Yves Le Traon. Extracting statistical graph features for accurate and efficient time series classification. In *21st International Conference on Extending Database Technology*, 2018.

[40] Chuang Liu, Wei-Xing Zhou, and Wei-Kang Yuan. Statistical properties of visibility graph of energy dissipation rates in three-dimensional fully developed turbulence. *Physica A: Statistical Mechanics and its Applications*, 389(13):2675–2681, 2010.

[41] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[42] Bartolo Luque, Lucas Lacasa, Fernando Ballesteros, and Jordi Luque. Horizontal visibility graphs: Exact results for random time series. *Physical Review E*, 80(4):046103, 2009.

[43] Norbert Marwan, Jonathan F Donges, Yong Zou, Reik V Donner, and Jürgen Kurths. Complex network approach for recurrence analysis of time series. *Physics Letters A*, 373 (46):4246–4254, 2009.

[44] Pablo Montero and José A. Vilar. TSclust: An R package for time series clustering. *Journal of Statistical Software*, 62(1):1–43, 2014.

[45] Manfred Mudelsee. *Climate time series analysis*. Springer, 2013.

[46] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2): 167–256, 2003.

[47] S original by Chris Fraley, U.Washington, Seattle. R port by Fritz Leisch at TU Wien; since 2003-12: Martin Maechler; fdGPH, fdSperio, etc by Valderio Reisen, and Artur Lemonte. *fracdiff: Fractionally differenced ARIMA aka ARFIMA(p,d,q) models*, 2012. R package version 1.4-2.

[48] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[49] John Paparrizos and Luis Gravano. Fast and accurate time-series clustering. *ACM Transactions on Database Systems (TODS)*, 42(2):8, 2017.

[50] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *International symposium on computer and information sciences*, pages 284–293. Springer, 2005.

[51] Meng-Cen Qian, Zhi-Qiang Jiang, and Wei-Xing Zhou. Universal and nonuniversal allometric scaling behaviors in the visibility graphs of world stock market indices. *Journal of Physics A: Mathematical and Theoretical*, 43(33):335002, 2010.

[52] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018.

[53] Pedro Ribeiro and Fernando Silva. G-tries: a data structure for storing and finding subgraphs. *Data Mining and Knowledge Discovery*, 28(2):337–377, 2014.

[54] Zhi-Gang Shao. Network analysis of human heartbeat dynamics. *Applied Physics Letters*, 96(7):073703, 2010.

[55] AH Shirazi, G Reza Jafari, J Davoudi, J Peinke, M Reza Rahimi Tabar, and Muhammad Sahimi. Mapping stochastic processes onto complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(07):07046, 2009.

[56] Robert H Shumway and David S Stoffer. *Time series analysis and its applications*. Springer, 2017.

[57] Isabel Silva, M Eduarda Silva, Isabel Pereira, and Nélia Silva. Replicated inar(1) processes. *Methodology and Computing in applied Probability*, 7(4):517–542, 2005.

[58] Supriya Supriya, Siuly Siuly, Hua Wang, Jinli Cao, and Yanchun Zhang. Weighted visibility graph with complex network features in the detection of epilepsy. *IEEE Access*, 4:6554–6566, 2016.

[59] Luciano Telesca and Michele Lovallo. Analysis of seismic sequences by using the method of visibility graph. *EPL (Europhysics Letters)*, 97(5):50002, 2012.

[60] Howell Tong. Threshold models in time series analysis—30 years on. *Statistics and its Interface*, 4(2):107–118, 2011.

[61] Ruey S Tsay. *Analysis of financial time series*, volume 543. John Wiley & Sons, 2005.

[62] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of machine learning research*, 15(1):3221–3245, 2014.

[63] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *International Work-Conference on Artificial Neural Networks*, pages 758–770. Springer, 2005.

[64] Xiaozhe Wang, Kate Smith, and Rob Hyndman. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, 13(3):335–364, 2006.

[65] Vitali Witowski and Dr. Ronja Foraita. *HMMpa: Analysing accelerometer data using hidden Markov models*, 2014. R package version 1.0.

[66] Diethelm Wuertz, Tobias Setz, and Yohan Chalabi. *timeSeries: Rmetrics - Financial Time Series Objects*, 2017. R package version 3042.102.

[67] Diethelm Wuertz, Tobias Setz, Yohan Chalabi, Chris Boudt, Pierre Chausse, and Michal Miklovac. *fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling*, 2017. R package version 3042.83.

[68] Xiaoke Xu, Jie Zhang, and Michael Small. Superfamily phenomena and motifs of networks induced from time series. *Proceedings of the National Academy of Sciences*, 105(50):19601–19605, 2008.

[69] Yue Yang, Jianbo Wang, Huijie Yang, and Jingshi Mang. Visibility graph approach to exchange rate series. *Physica A: Statistical Mechanics and its Applications*, 388(20):4431–4437, 2009.

[70] Jie Zhang and Michael Small. Complex network from pseudoperiodic time series: Topology versus dynamics. *Physical review letters*, 96(23):238701, 2006.

[71] Guohun Zhu, Yan Li, and Peng Paul Wen. Analysis and classification of sleep stages based on difference visibility graphs from a single-channel eeg signal. *IEEE journal of biomedical and health informatics*, 18(6):1813–1821, 2014.

[72] Enyu Zhuang, Michael Small, and Gang Feng. Time series analysis of the developed financial markets' integration using visibility graphs. *Physica A: Statistical Mechanics and its Applications*, 410:483–495, 2014.

[73] Walter Zucchini, Iain L MacDonald, and Roland Langrock. *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC, 2016.